

10080-00A0
APRIL 1973

NAKED MINI LSI/ALPHA LSI ENGINEERING SPECIFICATION



COMPUTER AUTOMATION
the NAKED MINI company

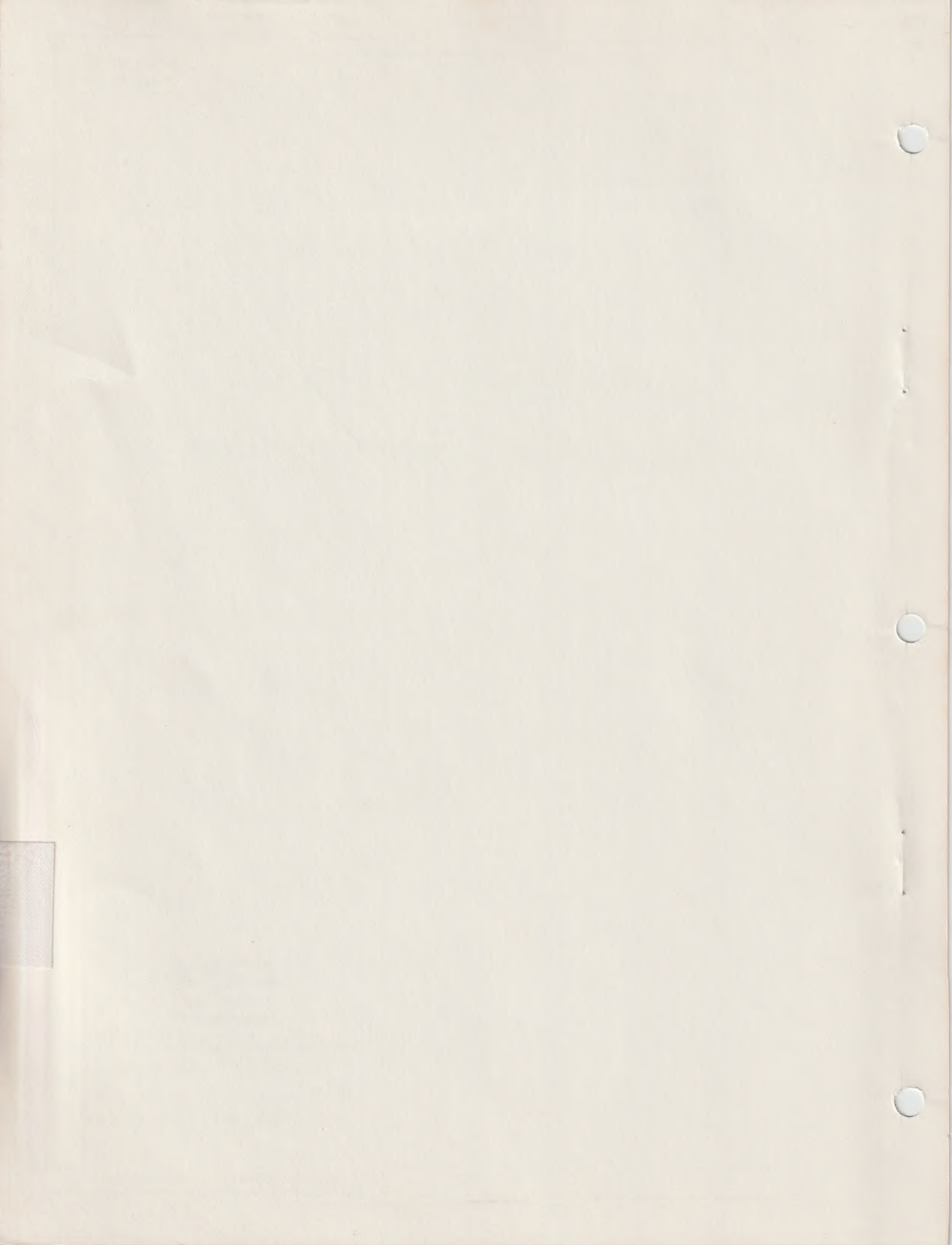
CAI Limited,
95a High Street,
Rickmansworth, Herts, WD3 1RB
Tel: Rickmansworth 79704/7 Telex: 922854



COMPUTER AUTOMATION, INC.
the NAKED MINI company

18651 Von Karman, Irvine, Calif. 92664
tel. 714-833-8830 TWX 910-595-1767

COPYRIGHT 1973, COMPUTER AUTOMATION, INC.



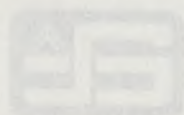
NAKED MINI LSI/ALPHA LSI ENGINEERING SPECIFICATION



COMPUTER AUTOMATION, INC.
the NAKED MINI company
18651 Von Karman, Irvine, Calif. 92664
tel. 714-833-8830 TWX 910-595-1767

10000-0000
APRIL 1973

NAKED MINI LSI\ALPHA LSI ENGINEERING SPECIFICATION



COMPUTER AUTOMATION, INC.
The NAKED MINI Company
1862 Van Rensselaer Blvd. Fort Worth, Texas 76102
Tel. 817-833-8830 TWX 910-2800

COPYRIGHT 1973 COMPUTER AUTOMATION, INC.



TABLE OF CONTENTS

Section		Page
Section 1. INTRODUCTION		
1.1	SCOPE.....	1-1
1.2	RELATED DOCUMENTATION.....	1-1
Section 2. CENTRAL PROCESSING UNIT		
2.1	GENERAL.....	2-1
2.1.1	General Characteristics.....	2-1
2.2	ORGANIZATION.....	2-2
2.2.1	Processor.....	2-2
2.2.2	Machine Registers.....	2-2
2.2.3	Processor.....	2-4
2.2.4	Internal Bussing.....	2-4
2.3	WORD AND INSTRUCTION FORMATS.....	2-4
2.4	INSTRUCTION REPERTOIRE AND EXECUTION TIMES.....	2-4
2.5	UNDEFINED INSTRUCTIONS.....	2-4
2.6	MEMORY ADDRESSING MODES.....	2-5
2.7	MACHINE TIMING.....	2-5
Section 3. INSTRUCTION SET		
3.1	INTRODUCTION.....	3-1
3.1.1	General.....	3-1
3.1.2	Arithmetic Operations and Overflow.....	3-1
3.2	MEMORY REFERENCE INSTRUCTIONS.....	3-2
3.2.1	Word Mode Operations and Instruction Format.....	3-2
3.2.1.1	Word Mode Direct Addressing.....	3-2
3.2.1.2	Word Mode Indirect Addressing.....	3-2
3.2.1.3	Word Mode Direct Indexed Addressing.....	3-3
3.2.1.4	Word Mode Indirect Post-Indexed Addressing.....	3-4
3.2.1.5	Word Mode Summary.....	3-4



TABLE OF CONTENTS (Cont'd)

Section		Page
3.2.2	Byte Mode Operations and Instruction Format.....	3-5
3.2.2.1	Byte Mode Direct Addressing.....	3-5
3.2.2.2	Byte Mode Indirect Addressing.....	3-5
3.2.2.3	Byte Mode Direct Indexed Addressing.....	3-7
3.2.2.4	Byte Mode Indirect Post-Indexed Addressing.....	3-7
3.2.2.5	Byte Mode Summary.....	3-7
3.2.3	Arithmetic Memory Reference Instructions.....	3-8
3.2.4	Logical Memory Reference Instructions.....	3-8
3.2.5	Data Transfer Memory Reference Instructions.....	3-8
3.2.6	Program Transfer Memory Reference Instructions.....	3-9
3.3	DOUBLE-WORD MEMORY REFERENCE INSTRUCTIONS.....	3-11
3.3.1	Format.....	3-11
3.3.2	Instructions.....	3-11
3.4	BYTE IMMEDIATE INSTRUCTIONS.....	3-14
3.4.1	Format.....	3-14
3.4.2	Instructions.....	3-14
3.5	CONDITIONAL JUMP INSTRUCTIONS.....	3-15
3.5.1	Format.....	3-15
3.5.2	Microcoding.....	3-15
3.5.3	Arithmetic Conditional Jump Instructions.....	3-16
3.5.4	Control Conditional Jump Instructions.....	3-17
3.6	SHIFT INSTRUCTIONS.....	3-17
3.6.1	Instruction Format.....	3-17
3.6.2	Arithmetic Shift Instructions.....	3-18
3.6.3	Logical Shift Instructions.....	3-19
3.6.4	Rotate Shift Instructions.....	3-20
3.6.5	Double Register (Long) Shift Instructions.....	3-21
3.6.6	Double Register (Long) Rotate Shift Instructions.....	3-21
3.7	REGISTER CHANGE INSTRUCTIONS.....	3-22
3.7.1	Format.....	3-22
3.7.2	A Register Change Instructions.....	3-22
3.7.3	X Register Change Instructions.....	3-23
3.7.4	OV Register Change Instructions.....	3-23
3.7.5	Multi-Register Change Instructions.....	3-24
3.7.6	Console Register Instructions.....	3-25



TABLE OF CONTENTS (Cont'd)

Section		Page
3.8	CONTROL INSTRUCTIONS.....	3-26
3.8.1	Format.....	3-26
3.8.2	Processor Control Instructions.....	3-26
3.8.3	Mode Control Instructions.....	3-26
3.8.4	Status Control Instructions.....	3-26
3.8.5	Interrupt Control Instructions.....	3-28
3.9	INPUT/OUTPUT INSTRUCTIONS.....	3-28
3.9.1	Control Input/Output Instructions.....	3-29
3.9.1.1	Sense Instructions.....	3-29
3.9.1.2	Select Instructions.....	3-30
3.9.2	Word Input/Output Instructions.....	3-30
3.9.2.1	Unconditional Word Input/Output Instructions.....	3-30
3.9.2.2	Conditional Word Input/Output Instructions.....	3-31
3.9.3	Byte Input Instructions.....	3-32
3.9.3.1	Unconditional Byte Input Instructions.....	3-32
3.9.3.2	Conditional Byte Input Instructions.....	3-33
3.9.4	Block Input/Output Instructions.....	3-34
3.9.5	Automatic Input/Output Instructions.....	3-35

Section 4. MEMORY

4.1	GENERAL.....	4-1
4.2	BASIC MEMORY.....	4-1
4.3	MEMORY EXPANSION.....	4-1
4.4	INTERCHANGEABILITY.....	4-2
4.5	MEMORY-TO-PROCESSOR INTERFACE.....	4-2
4.6	MEMORY SYSTEM ACCESS TIME.....	4-2

Section 5. INPUT/OUTPUT, MEMORY INTERFACE

5.1	GENERAL.....	5-1
5.2	EXTERNAL I/O Bus Components.....	5-1
5.2.1	Address Bus (A).....	5-1
5.2.2	Data Bus (D).....	5-3



TABLE OF CONTENTS (Cont'd)

Section		Page
5.2.3	Control Bus (C)	5-3
5.2.3.1	I/O Commands	5-3
5.2.3.2	Utility Signals	5-4
5.2.3.3	Interrupt Signals	5-4
5.2.3.4	DMA Signals	5-5
5.2.3.4.1	DMA Controller-to-Processor Interface	5-6
5.2.3.4.2	DMA Controller-to-Memory Interface	5-6
5.3	I/O TRANSFER TIMING	5-7
5.3.1	Input Timing	5-8
5.3.2	Output Timing	5-8
5.3.3	Sense Timing	5-8
5.3.4	Control Timing	5-8
5.4	INTERRUPTS	5-9
5.4.1	Interrupt Enabling and Disabling	5-9
5.4.2	Interrupt Masking	5-10
5.4.3	Reserved Memory Locations	5-10
5.4.4	Interrupt Address Offset	5-11
5.4.5	Interrupt Priority	5-12
5.4.6	Interrupt Timing	5-13
5.4.6.1	IL1 and IL2 Timing	5-13
5.4.6.2	IUR Timing	5-13
5.5	DIRECT MEMORY ACCESS (DMA)	5-14
5.5.1	Multiple DMA Controllers	5-15
5.5.2	DMA Controller Slots	5-15
5.5.3	Latency, Data Transfer Rate	5-15
5.5.4	Memory Overlap	5-15
5.6	ELECTRICAL CHARACTERISTICS	5-15
5.7	MOTHERBOARD ORGANIZATION	5-16
5.7.1	Interrupt Priority	5-19
5.7.2	Memory Bank Control, DMA Priority	5-19
5.7.3	CPU Power Supply Signals	5-19
5.8	EXPANSION	5-19
5.9	NAKED MINI 16 LSI EXTERNAL BUS REQUIREMENTS	5-20
5.10	TWO-MODULE OPTIONS	5-20



TABLE OF CONTENTS (Cont'd)

Section		Page
	Section 6. CONSOLE	
6.1	GENERAL.....	6-1
6.2	CONSOLE ORGANIZATION.....	6-1
6.3	FUNCTIONAL DESCRIPTION.....	6-2
6.4	PROGRAMMED OPERATIONS.....	6-7
6.4.1	Data Register Instructions.....	6-7
6.4.2	Program Status Word.....	6-7
6.4.3	Console Control Word.....	6-7
6.5	CONSOLE - PROCESSOR INTERFACE.....	6-8
6.6	PHYSICAL INTERCONNECT.....	6-10
	Section 7. OPTION CARD	
7.1	GENERAL.....	7-1
7.2	TELETYPE/CRT CONTROLLER.....	7-1
7.2.1	Functional Description.....	7-1
7.2.2	Transfer Rates.....	7-1
7.2.3	External Logic Levels.....	7-2
7.2.4	Interrupt Locations.....	7-2
7.2.5	Alternate Interrupt Locations.....	7-3
7.2.6	Interrupt Location Offset.....	7-3
7.2.7	I/O Instructions.....	7-3
7.2.7.1	Select Commands.....	7-3
7.2.7.2	Sense Commands.....	7-3
7.2.7.3	Input Commands.....	7-4
7.2.7.4	Output Commands.....	7-4
7.3	REAL TIME CLOCK.....	7-4
7.3.1	Functional Description.....	7-4
7.3.2	Clock Sources.....	7-5
7.3.3	Interrupts.....	7-5
7.3.4	I/O Instructions.....	7-5



TABLE OF CONTENTS (Cont'd)

Section		Page
7.4	AUTOLOAD.....	7-6
7.4.1	Functional Description.....	7-6
7.4.2	The Autoload Sequence.....	7-6
7.4.3	Programmed ROM Entry.....	7-6
7.4.4	Standard Autoload Program.....	7-7
7.5	OPTION JUMPER CONNECTOR.....	7-7

Section 8. POWER SUPPLY

8.1	GENERAL.....	8-1
8-2	INPUT POWER.....	8-1
8-3	OUTPUT POWER.....	8-1
8.4	TRANSIENT RESPONSE.....	8-1
8.5	OVER CURRENT AND THERMAL OVERLOAD PROTECTION.....	8-2
8.6	OVER VOLTAGE AND REVERSE VOLTAGE PROTECTION.....	8-2
8.7	LOGIC SIGNALS.....	8-2
8.8	POWERFAIL DETECTION AND SEQUENCING.....	8-3
8.8.1	Up Sequence.....	8-3
8.8.2	Down Sequence.....	8-3
8.9	MULTIPLE COMPUTER OPERATION.....	8-3
8.10	STANDARD POWER REQUIREMENTS.....	8-4

Section 9. PHYSICAL SPECIFICATION

9.1	INTRODUCTION.....	9-1
9.2	REFERENCE DESIGNATION SYSTEM.....	9-1
9.3	ALPHA 16 LSI PACKAGING CONCEPT.....	9-3



TABLE OF CONTENTS (Cont'd)

Section		Page
9.4	CHASSIS CONCEPT	9-3
9.5	MOTHERBOARD ASSEMBLY CONCEPT	9-5
9.6	FAN HOUSING ASSEMBLY CONCEPT	9-5
9.7	POWER SUPPLY ASSEMBLY CONCEPT	9-5
9.8	CONSOLE ASSEMBLY CONCEPT	9-5
9.9	CARD MODULE PRINTED CIRCUIT BOARD MECHANICAL SPECIFICATION	9-6
9-10	ENVIRONMENTAL SPECIFICATIONS	9-9

Appendix A. INSTRUCTION SET IN NUMERICAL ORDER

LIST OF ILLUSTRATIONS

Figure		Page
2-1	Processor Organization	2-3
3-1	Word Mode Memory Reference Instruction Format	3-3
3-2	Word Mode Addressing Summary	3-4
3-3	Byte Mode Memory Reference Instruction Format	3-6
3-4	Byte Mode Addressing Summary	3-7
3-5	Double-Word Memory Reference Format	3-11
3-6	Divide	3-12
3-7	Multiply and Add	3-13
3-8	NRM Shift Path	3-13
3-9	Immediate Instruction Format	3-14
3-10	JOC Jump On Condition Format	3-16
3-11	Single Register Shift Format	3-17
3-12	Double Register (Long) Shift Format	3-18
3-13	Arithmetic Right Shift	3-18
3-14	Arithmetic Left Shift	3-18
3-15	Rotate Right	3-19
3-16	Rotate Left	3-19



LIST OF ILLUSTRATIONS (Cont'd)

Figure		Page
3-17	Rotate Right.....	3-20
3-18	Rotate Left.....	3-20
3-19	Long Left Shift.....	3-21
3-20	Long Right Shift.....	3-21
3-21	Long Rotate Right.....	3-21
3-22	Long Rotate Left.....	3-22
3-23	Register Change Format.....	3-22
3-24	Control Format.....	3-26
3-25	Computer Status Word Format.....	3-27
3-26	Single Word Input/Output Instruction Format.....	3-29
3-27	Block Input/Output Instruction Format.....	3-34
3-28	Automatic Input/Output Instruction Format.....	3-35
3-29	In-line Auto I/O Instruction Sequence.....	3-36
3-30	Interrupt Location Auto I/O Instruction Sequence.....	3-37
5-1	I/O Bus Components.....	5-2
5-2	I/O Transfer Timing.....	5-8
5-3	Interrupt Transfer Timing.....	5-14
5-4	ALPHA 16 LSI Motherboard Slot Organization (Rear View).....	5-18
6-1	ALPHA 16 LSI Console.....	6-3
9-1	ALPHA 16 LSI Reference Designator System.....	9-2
9-2	ALPHA 16 LSI Outline and Mounting.....	9-4
9-3	Full Board Design Guide.....	9-7
9-4	Half Board Design Guide.....	9-8
9-5	ALPHA 16 LSI Ventilation Systems.....	9-10

LIST OF TABLES

Table		Page
5-1	External I/O Bus Load, Drive and Termination Summary.....	5-17
5-2	Slot A100 Special Signals.....	5-18
6-1	Console Switches and Indicators.....	6-2



Section 1

INTRODUCTION

1.1 SCOPE

This specification defines the NAKED MINI 16 LSI and ALPHA 16 LSI computers. The ALPHA 16 LSI computer consists of:

1. A NAKED MINI 16 LSI plug-in module consisting of a MOS/LSI Processor and up to 8192 sixteen bit words of memory,
2. A computer option module which contains the Teletype/CRT interface, Real Time Clock, and Autoload options,
3. A motherboard which interconnects the Processor with additional I/O and memory modules, the power supply, the operator's console and expansion chassis,
4. A system power supply module,
5. An operator's console,
6. A metal chassis for packaging all of the above.

The option card (item 2 above) mounts in piggy-back fashion to the NAKED MINI 16 LSI computer to form a stand-alone computer system. In this configuration, items 3, 4, 5 and 6 are not supplied.

1.2 RELATED DOCUMENTATION

Detailed user-oriented information relating to the ALPHA LSI computer system is available in the following documents:

<u>Document No.</u>	<u>Description</u>
91-10079-00A0	NAKED MINI/ALPHA 16 LSI Interface Design Guide
91-10077-00A0	NAKED MINI/ALPHA 16 LSI Programming Reference Manual
91-10081-00A0	NAKED MINI/ALPHA 16 LSI Computer Reference Manual



Section 2

CENTRAL PROCESSING UNIT

2.1 GENERAL

This section provides an overview of the NAKED MINI 16 LSI computer Central Processing Unit (Processor). Both the electrical and physical organization of the Processor are described.

2.1.1 General Characteristics

Some of the significant features of the Processor are summarized below:

1. Binary 2's complement arithmetic
2. 16-bit parallel processing
3. 8- or 16-bit word format
4. Seven 16-bit hardware registers and one 8-bit Status register
5. Direct, indexed, relative and multilevel indirect addressing (word mode and byte mode)
6. Direct Memory Channels (interrupt driven automatic I/O)
7. Direct Memory Access (DMA)
8. Parallel I/O
9. Five interrupt levels
10. Power fail protect/restart (optional)
11. Microprogrammed control unit
12. Hardware multiply, divide



2.2 ORGANIZATION (Figure 2-1)

2.2.1 Processor

The Processor is implemented completely within two MOS/LSI chip types, excluding the four phase timing generator and the Processor-to-external bus interface (MOS-TTL conversion).

The two MOS/LSI chip types are the Processor Chip (P-chip) and Control Chip (C-chip). Both are P-channel Silicon gate devices.

The P-chip contains the arithmetic unit with carry lookahead, register file, status flip-flops (including overflow), and chip interface logic, organized on a 4-bit slice. Four P-chips are required to complete a 16-bit "data loop".

The C-chip contains a macro-instruction register with 4-bit down-count capability, a status inhibit counter, interrupt logic, a programmable logic array (PLA) and PLA/micro-instruction conversion logic. The PLA is the microprogram store of the Processor. The conversion logic performs some control functions in "hardwired" logic to simplify the microprogram-oriented structure of the machine.

Each C-chip implements a subset of the instructions of the Processor. Three C-chips are required to fully implement the machine. At any given time, only one C-chip is "active", issuing micro-instructions to other chips in the system.

Interrupts and console operations are, like macro-instructions, microprogrammed.

2.2.2 Machine Registers

The machine includes the following registers:

A	Accumulator; 16-bits. Program accessible.
X	Index and secondary accumulator; 16-bits. Program accessible.
I	Instruction register; 16-bits.
P	Program counter; 16-bits.
W	} Working registers: all 26-bits.
T	
B	
OV	Overflow flip-flop; 1 bit. Program accessible.
BS	Bit Save or Carry flip-flop; 1 bit.
BM	Byte Mode flip-flop; 1 bit. Program accessible.
MIR	Macro-Instruction register; 16-bits.
SIR	Status Inhibit register; 3-bit down counter. Program loaded.
IEF	Interrupt Enable flip-flop; 1 bit. Program accessible.
PEF	Power Fail Interrupt Enable flip-flop; 1 bit. Program loaded.
CEF	Console Interrupt Enable flip-flop; 1 bit. Program loaded.

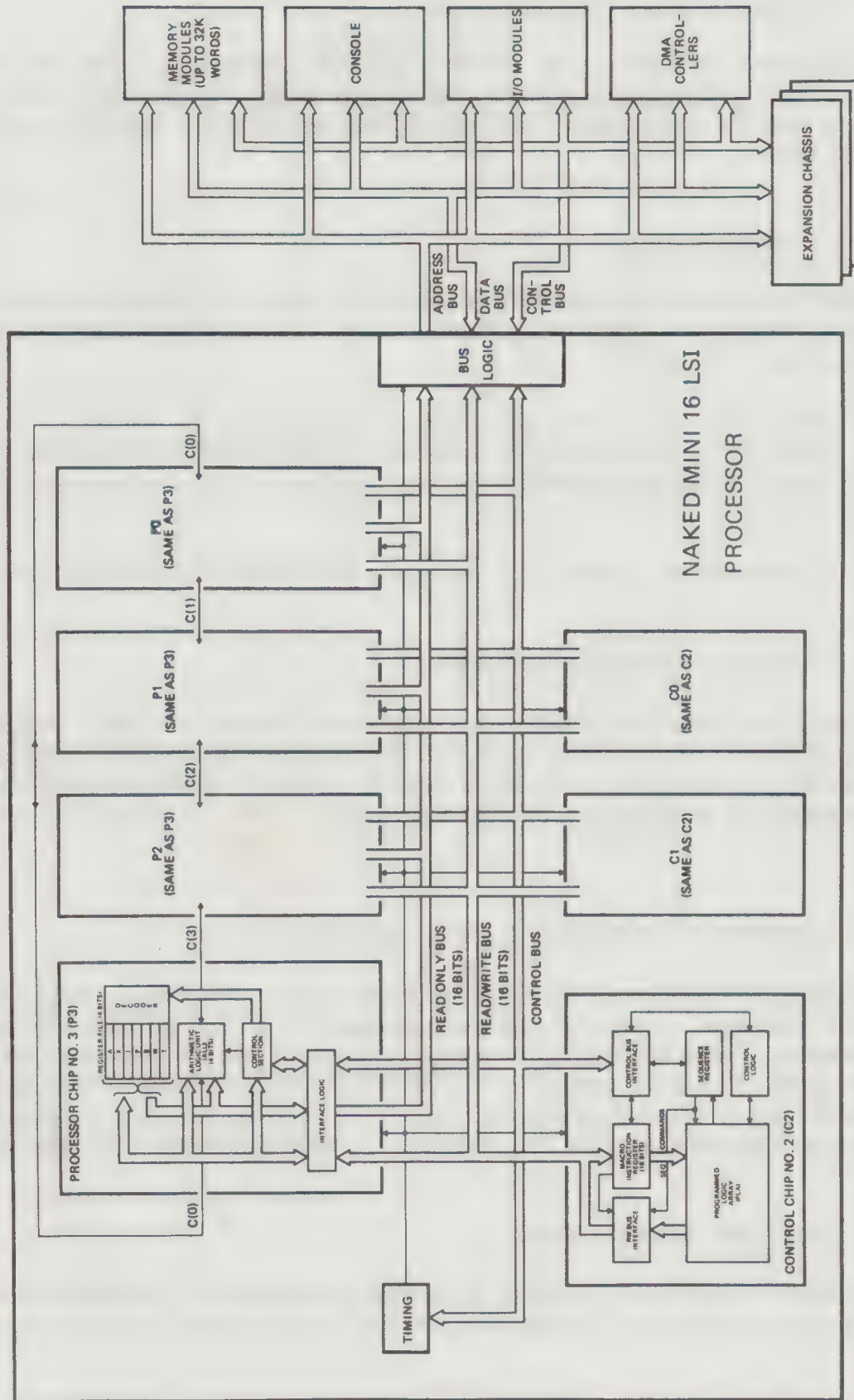


Figure 2-1. Processor Organization



2.2.3 Processor External Bus Interface

The Processor interfaces to the external bus via a network of sense amp tri-state transmitters and tri-state receivers with flip-flop storage. The external bus is used for both memory and I/O operations. The external bus consists of a data bus, address bus, and control bus.

2.2.4 Internal Bussing

Internal to the Processor, a two-bus scheme is utilized. The address bus (ROXX) emanates from the P-chips, four lines per chip. The data bus (RWXX) connects all C-chips and P-chips.

Instructions from the memory subsystem are transferred via the Data Bus and stored in the P-chips and C-chips. Data from/to memory utilizes the P-chips only. The Data Bus is time-shared for the transfer of microcontrol information between C-chips and P-chips.

Control information is issued from and gated into both C-chips and P-chips.

2.3 WORD AND INSTRUCTION FORMATS

One-word, two-word and three-word instruction formats are used. Single precision data formats are used, except for the double precision operands of multiply, divide and normalize, and byte operands of memory reference instructions executed when the computer is programmed into the byte mode. Refer to section 3 for more information.

2.4 INSTRUCTION REPERTOIRE AND EXECUTION TIMES

Each instruction is executed as a sequence of micro-instructions. One micro-cycle is 1.6 microseconds, except for phase-stretching which occurs in memory and I/O communication. Phase stretching for memory communication is dependent on memory access time. Phase stretching for I/O communication is preset to 600 nanoseconds (for a micro-cycle of 2.2 microseconds) to allow for address and data transmission time and sense response from the I/O Controller. Refer to section 3 for more information.

2.5 UNDEFINED INSTRUCTIONS

Only those instructions described in section 3 are defined. Undefined instructions will execute as no-op's. Undefined op-codes are reserved for future expansion.



2.6 MEMORY ADDRESSING MODES

The memory reference instructions access operands in memory either in the Word Mode or Byte Mode, depending on the state of the Byte Mode flip-flop (controlled by two dedicated control instructions). Within each mode, eight variations are permitted. Refer to section 3 for more details.

2.7 MACHINE TIMING

The basic Processor clock is a 20 MHz oscillator. The basic machine cycle time is 1.6 microseconds, comprised of eight time slots of 200 nanoseconds each. Four time slots correspond to the four clock phases of the system. Each clock phase is separated by a 200 nanosecond "guard band".

Also derived from the basic oscillator are 10 MHz and 1.0 MHz clocks for the I/O Subsystem.



Subscription price, \$5.00 per annum in advance.

Single copies, 15 cents. For those who wish to receive the Journal by mail, the subscription price must be paid in advance. The Journal is sent by mail to all subscribers, except those who live in the same city as the publisher, who receive it by express.

Entered as Second-Class Matter, May 2, 1879.

Postpaid. The Journal is sent by mail to all subscribers, except those who live in the same city as the publisher, who receive it by express. The subscription price must be paid in advance. The Journal is sent by mail to all subscribers, except those who live in the same city as the publisher, who receive it by express. The subscription price must be paid in advance.





Section 3

INSTRUCTION SET

3.1 INTRODUCTION

3.1.1 General

The instruction set of the ALPHA LSI computer is divided into seven classes:

1. Memory Reference
2. Byte Immediate
3. Conditional Jump
4. Shift
5. Register Change
6. Control
7. Input/Output

The Memory Reference class processes data in either Word Mode or Byte Mode. Because of the significance of these two modes of processing, the Memory Reference class is divided into two subclasses. The subclasses are Memory Reference: Word Mode, and Memory Reference: Byte Mode.

Each instruction class and subclass is treated separately in this section. The general format of each class is explained along with any special considerations.

3.1.2 Arithmetic Operations and Overflow

The ALPHA LSI computer performs two's-complement arithmetic. All additions and subtractions are performed on full 16-bit values. Thus, addition operations involving byte values place the 8-bit data in the least significant 8 bits of the adder and set the most significant 8 bits to zero (e.g., AXI : 50 would add : 0050 to the 16-bit X register). Subtraction operations involving byte values similarly obtain a 16-bit two's complement of the data (e.g., SXI : 50 would add : FFB0 to the 16-bit X register).



Arithmetic overflow occurs when the result of an arithmetic operation exceeds the range -32768 through +32767. Specifically, this involves the carry from bit 14 to bit 15 of the adder, and the carry out of bit 15. If the carries are not equal (0 and 1, or 1 and 0), an arithmetic overflow has occurred and the OV (overflow) indicator is set.

3.2 MEMORY REFERENCE INSTRUCTIONS

3.2.1 Word Mode Operations and Instruction Format

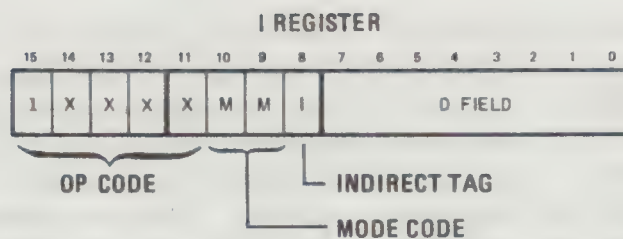
Word mode memory reference operations access full 16-bit memory operands. The default mode of the computer is the Word Mode - i.e., when no mode control instruction has been executed, the computer is in the Word Mode. SWM is the mode control instruction which places the computer in the Word Mode. In addition, the SIN, SIA and SIX instructions force the computer into the Word Mode. The SIN instruction forces the Word Mode for the number of succeeding instructions specified by its associated operand. The SIA and SIX instructions unconditionally force the Word Mode. The format for the Word Mode Memory Reference instructions is shown in Figure 3-1.

3.2.1.1 Word Mode Direct Addressing

Word Mode direct addressing allows any memory reference instruction to access the first 256 words of memory (the base page/scratchpad area) as well as 511 memory locations about the instruction itself (relative to P). Relative to P forward addressing includes 256 words forward (toward higher memory) of the instruction and relative to P backwards addressing includes the instruction itself and 255 memory locations backward from the instruction.

3.2.1.2 Word Mode Indirect Addressing

Word Mode indirect addressing allows any memory reference instruction to access any memory location through an address pointer in the first 256 words of memory (the base page/scratchpad area) or an address pointer in the 511 memory locations about the instruction itself (relative to P). Relative to P forward indirect addressing allows the address pointer to reside in memory locations as many as 256 words forward (toward higher memory) of the instruction and relative to P backwards indirect addressing allows the address pointer to be in any memory location 255 words or less prior to the instruction. Multi-level indirect addressing is accomplished by accessing address pointers in which the most significant bit (bit 15) is set. The memory operand is not accessed until an address pointer with the most significant bit reset (=0) is encountered.



Field	Bits	Definition
Op Code	11-15	Operation Code. Defines the specific instructions to be executed.
Mode Code (M Field)	9, 10	Used in conjunction with the Indirect Tag to define the memory addressing mode to be used: M = 00 Scratchpad M = 01 Relative to P, Forward M = 10 Indexed M = 11 Relative to P, Backward
Indirect Tag (I)	8	Specifies Direct or Indirect addressing. I = 0 Direct. M specifies operand address. I = 1 Indirect. M specifies address pointer address. Address Pointer specifies operand address, or the address of another Address Pointer.
D Field	0-7	Base value used to form operand or address pointer address.

Figure 3-1. Word Mode Memory Reference Instruction Format

3.2.1.3 Word Mode Direct Indexed Addressing

Word Mode direct indexed addressing allows any memory reference instruction to access memory locations by summing the contents of the X register and any offset value in the range 0 through 255. The offset value is defined by the D field in the instruction.

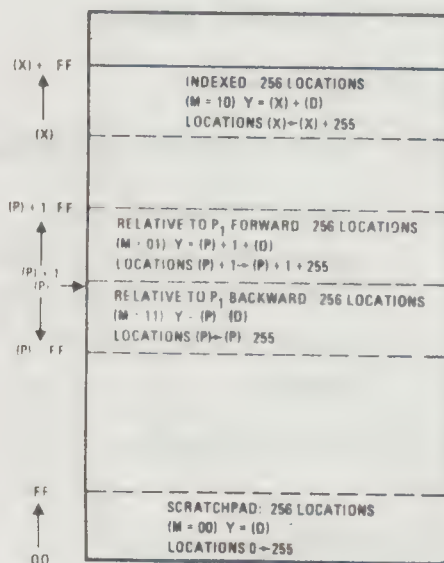


3.2.1.4 Word Mode Indirect Post-Indexed Addressing

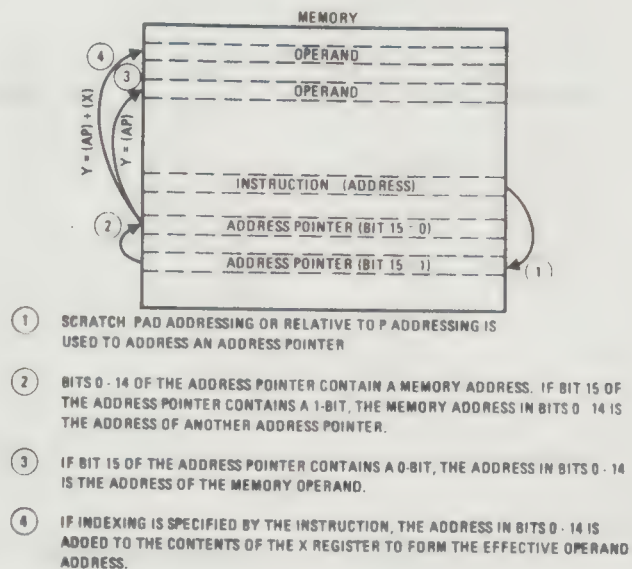
Word Mode indirect post-indexed addressing allows any memory reference instruction to access memory locations by summing the contents of the X register and the contents of an address pointer in the base page/scratchpad area (the first 256 memory locations). If the most significant bit of the address pointer is set, it contains the address of another address pointer, which in turn may contain the address of another pointer, and so forth. When an address pointer with the most significant bit (bit 15) set to zero is found, the contents of the X register are added to it to form the effective memory address. The memory operand is then accessed.

3.2.1.5 Word Mode Summary

A summary of Word Mode addressing is shown in Figure 3-2.



Direct Addressing



Indirect Addressing

Figure 3-2. Word Mode Addressing Summary



3.2.2 Byte Mode Operations and Instruction Format

Byte Mode memory reference operations access 8-bit byte operands. The Byte Mode is established by execution of the Set Byte Mode (SBM) instruction. Byte Mode memory reference operation is inhibited (the computer is forced into the Word Mode) by execution of the SIN, SWM, SIA and SIX instructions. The SIN instruction inhibits Byte Mode operations for the number of succeeding instructions specified by its associated operand. The SWM, SIA and SIX instructions unconditionally force the computer into the Word Mode. The format for the Byte Mode memory reference instructions is shown below in Figure 3-3.

All (16-bit) byte address pointers consist of fifteen bits of word address in the most significant 15 bits. The least significant bit (bit 0) specifies the most significant 8 bits (=0) or the least significant 8 bits (=1) of the word. Only one level of byte memory reference indirect addressing - specified in the instruction itself - is possible. Byte operands affecting the register are always right-justified - i.e., bytes cannot be loaded into, added to or stored from the most significant 8 bits of the A and X registers.

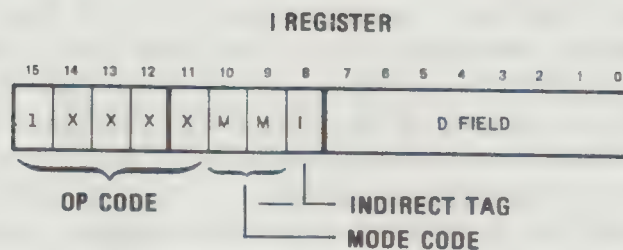
The IMS, MPY, DVD, NRM, JMP and JST instructions are not affected by the Byte Mode. They always use full 16-bit word operands.

3.2.2.1 Byte Mode Direct Addressing

Byte Mode direct addressing allows any byte memory reference instruction to access the first 256 bytes (128 words) of memory as well as 512 byte locations ahead (toward high-order memory) of the instruction itself.

3.2.2.2 Byte Mode Indirect Addressing

Byte Mode indirect addressing allows any byte memory reference instruction to access any byte location through a byte address pointer in the first 256 words of memory (the base page/scratchpad area) or a byte address pointer in the 511 memory locations about the instruction itself (relative to P). Relative to P forward indirect addressing allows the byte address pointer to reside in memory locations as many as 256 words forward (toward higher memory) of the instruction and relative to P backwards indirect addressing allows the byte address pointer to be in any memory location 255 words or less backward from the instruction. Since a byte address pointer utilizes all 16 bits to specify a given byte location, indirect byte addressing is limited to one level.



Field	Bits	Definition
OpCode	11-15	Operation Code. Defines the specific instruction.
M Field (Mode)	9, 10	Mode Code. Used in conjunction with the Indirect Tag to define the memory addressing mode to be used.
I Bit	8	Indirect Tag. Specifies direct or indirect addressing.
D Field	0-7	Address Field. Base address used to form byte operand address or address pointer address.

Addressing Modes:

MM I

Direct Addressing:

00	0	Scratchpad	$Y(\text{byte})=(D)$
01	0	Relative to P, Forward; Byte 0	$Y(\text{word})=(P)+1+(D)$
11	0	Relative to P, Forward; Byte 1	$Y(\text{word})=(P)+1+(D)$
10	0	Indexed	$Y(\text{byte})=(D)+(X)$

Indirect Addressing:

00	1	AP in Scratchpad	$AP(\text{word})=(D), Y(\text{byte})=(AP)$
01	1	AP Relative to P, Forward	$AP(\text{word})=(P)+1+(D), Y(\text{byte})=(AP)$
11	1	AP Relative to P, Backward	$AP(\text{word})=(P)-(D), Y(\text{byte})=(AP)$
10	1	AP in Scratchpad, Indexed	$AP(\text{word})=(D), Y(\text{byte})=(AP)+(X)$

Figure 3-3. Byte Mode Memory Reference Instruction Format



3.2.2.3 Byte Mode Direct Indexed Addressing

Byte Mode direct indexed addressing allows any byte memory reference instruction to access byte locations by summing the contents of the X register and any base value in the range 0 through 255. The base value is defined by the D field of the instruction.

3.2.2.4 Byte Mode Indirect Post-Indexed Addressing

Byte Mode indirect post-indexed addressing allows any byte memory reference instruction to access byte locations by summing the contents of the X register and the contents of a byte address pointer in the base page/scratchpad area (the first 256 memory locations).

3.2.2.5 Byte Mode Summary

A summary of Byte Mode addressing is shown in Figure 3-4.

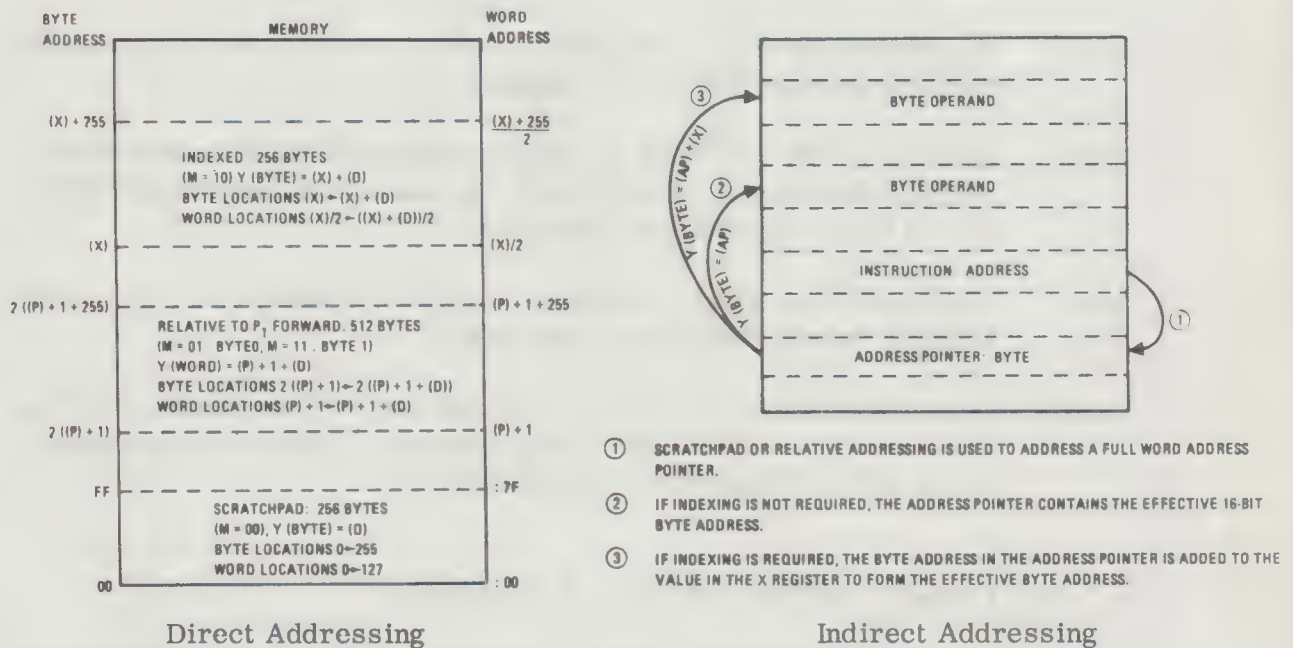


Figure 3-4. Byte Mode Addressing Summary



3.2.3 Arithmetic Memory Reference Instructions

- ADD** **ADD TO A.** Adds the contents of the effective memory location to the A register. OV is set if arithmetic overflow occurs.
- ADDB** **ADD BYTE TO A.** Adds the contents of the effective byte location to the A register. OV is set if arithmetic overflow occurs.
- SUB** **SUBTRACT FROM A.** Subtracts the contents of the effective memory location from the A register. OV is set if arithmetic overflow occurs.
- SUBB** **SUBTRACT BYTE FROM A.** Subtracts the contents of the effective byte location from the A register. OV is set if arithmetic overflow occurs.

3.2.4 Logical Memory Reference Instructions

- AND** **AND TO A.** Logically AND's the contents of the effective memory location with the A register.
- ANDB** **AND BYTE TO A.** Logically AND's the contents of the effective byte location with the A register. Resets the most significant 8 bits of the A register to zero.
- IOR** **INCLUSIVE OR TO A.** Inclusively OR's the contents of the effective memory location with the A register.
- IORB** **INCLUSIVE OR BYTE TO A.** Inclusively OR's the contents of the effective byte location with the A register. The most significant 8 bits of the A register are unchanged.
- XOR** **EXCLUSIVE OR TO A.** Exclusively OR's the contents of the effective memory location with the A register.
- XORB** **EXCLUSIVE OR BYTE TO A.** Exclusively OR's the contents of the effective byte location with the A register. The most significant 8 bits of the A register are unchanged.

3.2.5 Data Transfer Memory Reference Instructions

- EMA** **EXCHANGE MEMORY AND A.** Simultaneously stores the A register in the effective memory location and loads the contents of the effective memory location into the A register.



- EMAB EXCHANGE MEMORY BYTE AND A. Simultaneously stores the least significant 8 bits of the A register into the effective byte location and loads the contents of the effective byte location into the least significant 8 bits of the A register. Resets the most significant 8 bits of the A register to zero.
- LDA LOAD A. Loads the contents of the effective memory location into the A register.
- LDAB LOAD A BYTE. Loads the contents of the effective byte location into the least significant 8 bits of the A register. Resets the most significant 8 bits of the A register to zero.
- LDX LOAD X. Loads the contents of the effective memory location into the X register.
- LDXB LOAD X BYTE. Loads the contents of the effective byte location into the least significant 8 bits of the X register. Resets the most significant 8 bits of X register to zero.
- STA STORE A. Stores the A register in the effective memory location.
- STAB STORE A BYTE. Stores the least significant 8 bits of the A register in the effective byte location.
- STX STORE X. Stores the X register in the effective memory location.
- STXB STORE X BYTE. Stores the least significant 8 bits of the X register in the effective byte location.

3.2.6 Program Transfer Memory Reference Instructions

- CMS COMPARE AND SKIP IF HIGH OR EQUAL. Compares the contents of the effective memory location with the A register. If the A register is greater than the contents of the effective memory location, a one-word skip occurs. If the A register is equal to the contents of the effective memory location, a two-word skip occurs.
- CMSB COMPARE BYTE AND SKIP IF HIGH OR EQUAL. Compares the contents of the effective byte location with the A register. If the A register is greater than the contents of the effective byte location, a one-word skip occurs. If the A register is equal to the contents of the effective byte location, a two-word skip occurs. All 16 bits of the A register are compared to the contents of the effective byte location, so the most significant 8 bits of A register should be zeros.



IMS INCREMENT MEMORY AND SKIP ON ZERO RESULT. Contents of the effective memory location are incremented by one. If the increment causes the result to become zero, a one-word skip occurs. OV is set if arithmetic overflow occurs.

NOTE

IMS is often used as an interrupt instruction.

JMP JUMP UNCONDITIONAL. The P counter is loaded with the value of the effective memory location, causing an unconditional branch to that address.

JST JUMP AND STORE. The contents of the P counter are stored in the effective memory location and the P counter is then loaded with the effective memory location plus one.

NOTE

JST is often used as an interrupt instruction

SCM SCAN MEMORY. Compares the A register with area of memory defined by the Address Pointer in Scratchpad (Base Address of Buffer - 1) and the contents of the X register (Buffer Length). If a match is found, the Scan is terminated and the next instruction in sequence is executed. The X register is decremented once for each word scanned. When a match is found, the X register contains the number of words remaining to be scanned. The remainder of the buffer can be scanned simply by executing the SCN instruction again. If a match is not found by the time X reaches zero, a one place skip occurs and the instruction terminates. This instruction operates under both Word and Byte modes, the Address Pointer in Scratchpad should be constructed accordingly.

NOTE

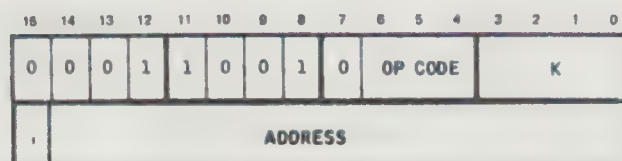
This instruction is interruptable. Upon completion of interrupt processing, SCM resumes operation at the point where the interrupt occurred.



3.3 DOUBLE-WORD MEMORY REFERENCE INSTRUCTIONS

3.3.1 Format

The Double-Word Memory Reference instructions require two consecutive words of memory and allow direct and indirect addressing. Indexed addressing is not allowed and is, in fact, not useful, since these instructions manipulate both the A and X registers. The format for Double-Word Memory Reference instructions is shown in Figure 3-5.



Op Code = 100 for NRM 16 through 31
 = 101 for NRM 0 through 15
 = 110 for MPY
 = 111 for DVD

I = Indirect Addressing
 1 = Indirect Address
 0 = Direct Address

K = Instruction Count

Figure 3-5. Double-Word Memory Reference Format

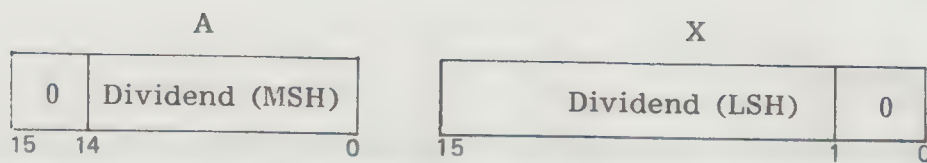
3.3.2 Instructions

DVD **DIVIDE**. Divides the contents of the A and X registers by the contents of the memory location addressed by the second word of the instruction. This address pointer may be direct or indirect.

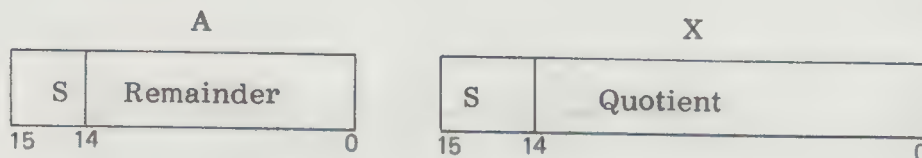
Prior to execution of the instruction, the A and X registers contain the signed 30 bit dividend (as shown in Figure 3-6), and the addressed memory location contains the signed full-word divisor. Both dividend and divisor must be positive.

The quotient is returned in the X register (sign plus 15 bits) and the remainder in the A register (sign plus 15 bits). OV is set if a divide fault occurs (divisor Most Significant Half (MSH) of the dividend), or else is returned in the same state as entered.

A full divide is performed if no instruction count is specified. Partial divides are executed according to the specified instruction count.



a. Registers Prior to DVD.



b. Registers after DVD

Figure 3-6. Divide

MPY MULTIPLY AND ADD. Multiplies the contents of the X register by the contents of the memory location addressed by the second word of the instruction and adds the contents of the A register to the product. The address pointer may be direct or indirect.

Prior to execution of the MPY instruction, the X register contains the signed full-word multiplicand, the addressed memory location contains the signed full-word multiplier, and the A-register contains the signed "offset" to be added. (Refer to Figure 3-7.) Multiplicand and offset must all be positive or zero.

The result is returned in the A and X registers (sign plus 30 bits). A full multiply and add is performed if no instruction count is specified. Partial multiplication is executed according to the specified instruction count.



In all cases OV will be reset (=0) at completion of a full multiply.



a. Registers Prior to MPY.



b. Register After MPY

Figure 3-7. Multiply and Add

NRM NORMALIZE A AND X. Contents of A and X registers are arithmetically shifted left (see Figure 3-8) until A15 is not equal to A14 or until the maximum shift count specified is reached. The exponent, addressed by the second word of the instruction is decremented once for each shift. The address of the exponent may be direct or indirect.

The NRM instruction treats the A and X register as a combined 31-bit plus sign register.

OV is reset (=0) if normalization occurs, otherwise it is returned unaltered. In either case, the exponent will be decremented once for each shift performed.

A full 31-bit normalize is performed if no instruction count is specified. Otherwise, the specified count will determine the maximum shifts performed.

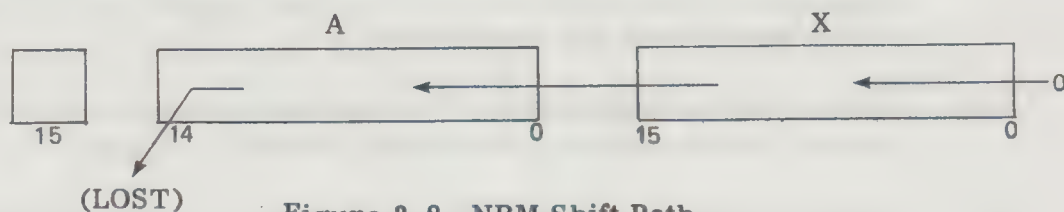


Figure 3-8. NRM Shift Path



3.4 BYTE IMMEDIATE INSTRUCTIONS

3.4.1 Format

Immediate instructions are similar to Memory Reference instructions in that they perform logical and arithmetic operations involving memory data and operating registers. The memory data, however, is stored within the immediate instruction itself rather than in a separate operand word or byte. The Immediate Instruction format is shown in Figure 3-9.

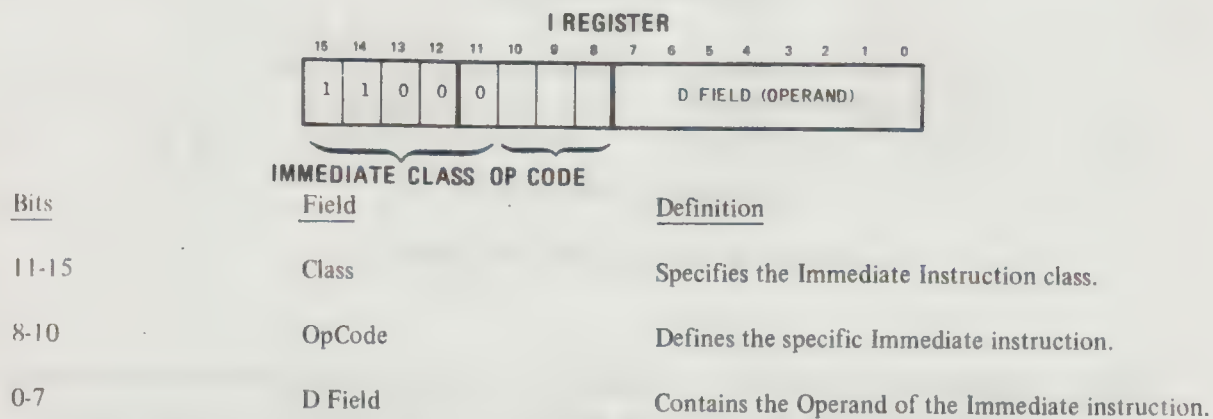


Figure 3-9. Immediate Instruction Format

3.4.2 Instructions

- AXI** ADD TO X IMMEDIATE. The operand is added to the contents of the X register. OV is set if arithmetic overflow occurs.
- CAI** COMPARE TO A IMMEDIATE. The operand is compared to the least significant 8 bits of the A register. If unequal, a skip of one place occurs. If equal, the next instruction in sequence is executed. The contents of A are not disturbed. The most significant 8 bits of A do not take part in the comparison.
- CXI** COMPARE TO X IMMEDIATE. The operand is compared to the least significant 8 bits of the X register. If unequal, a skip of one place occurs. If equal, the next instruction in sequence is executed. The contents of X are not disturbed. The most significant 8 bits of X do not take part in the comparison.
- LAM** LOAD A MINUS IMMEDIATE. The operand is negated (two's complemented) and loaded as a 16-bit word into the A register.
- LAP** LOAD A POSITIVE IMMEDIATE. The operand is loaded into the least significant 8 bits of the A register. The most significant 8 bits of A are set to zero.



- LXM LOAD X MINUS IMMEDIATE. The operand is negated (two's complemented) and loaded as a 16-bit word into the X register.
- LXP LOAD X POSITIVE IMMEDIATE. The operand is loaded into the least significant 8 bits of the X register. The most significant 8 bits of X are set to zero.
- SXI SUBTRACT FROM X IMMEDIATE. The operand is negated (two's complemented) and added as a 16-bit word to the X register. OV is set if arithmetic overflow occurs.

3.5 CONDITIONAL JUMP INSTRUCTIONS

3.5.1 Format

Conditional Jump instructions test conditions within the computer and perform program branches depending on the results of the test. A Jump occurs if the specified conditions are satisfied. All branches are direct and relative to the P register (location of the Conditional Jump instruction). The range of Conditional Jumps is:

Forward Jumps:	P+1 through P+64
Backward Jumps:	P through P-63

3.5.2 Microcoding

A general code, JOC, for Jump On Condition, is provided so the programmer can microcode jump conditions. There are five different conditions which may be tested either individually or in combination:

1. Sign of A (positive or negative)
2. Contents of A (zero or not zero)
3. Contents of X (zero or not zero)
4. Overflow Indicator (Set or Reset)
5. SENSE Indicator (on or off)

The conditions may be tested individually or in combination. Figure 3-10 shows the format for the JOC instruction:



Bits	Field	Definition																		
13-15	Class	Identifies the Conditional Jump Instruction Class																		
12	G	Test Group Indicator: G=1 for AND Group G=0 for OR Group																		
7-11	Conditions	Microcode of Test Conditions: <table> <tr> <th>Bit</th><th>AND Group</th><th>OR Group</th></tr> <tr> <td>7</td><td>A Positive</td><td>A Negative</td></tr> <tr> <td>8</td><td>A≠0</td><td>A=0</td></tr> <tr> <td>9</td><td>OV Reset</td><td>OV Set (Resets OV)</td></tr> <tr> <td>10</td><td>SS On</td><td>SS Off</td></tr> <tr> <td>11</td><td>X≠0</td><td>X=0</td></tr> </table>	Bit	AND Group	OR Group	7	A Positive	A Negative	8	A≠0	A=0	9	OV Reset	OV Set (Resets OV)	10	SS On	SS Off	11	X≠0	X=0
Bit	AND Group	OR Group																		
7	A Positive	A Negative																		
8	A≠0	A=0																		
9	OV Reset	OV Set (Resets OV)																		
10	SS On	SS Off																		
11	X≠0	X=0																		
6	R	Jump Direction: R=0 for Forward Jump R=1 for Backward Jump																		
0-5	D Field	Jump Distance (-63 to +64)																		

Figure 3-10. JOC Jump On Condition Format

JOC commands consist of two groups. The AND group and the OR group. The AND test group requires that all of the test conditions specified by bits 7 thru 11 of the instruction be true for the jump to take place. The OR group requires that any one or more of the test conditions specified be true if the jump is to take place.

The following Conditional Jump instructions are special cases of the general JOC instruction. Since they are more often utilized than the other conditional jumps, they have been given their own mnemonics.

3.5.3 Arithmetic Conditional Jump Instructions

- JAG JUMP IF A GREATER THAN ZERO. Jump occurs if the A register is greater than zero.
- JAL JUMP IF A LESS THAN ONE. Jump occurs if the A register is less than or equal to zero.



- JAM JUMP IF A MINUS. Jump occurs if the A register is less than zero ($A = 1$).
- JAN JUMP IF A NOT ZERO. Jump occurs if the A register is not zero.
- JAP JUMP IF A POSITIVE. Jump occurs if the A register is greater than or equal to zero ($A = 0$).
- JAZ JUMP IF A ZERO. Jump occurs if the A register is zero.
- JXN JUMP IF X NOT ZERO. Jump occurs if the X register is not zero.
- JXZ JUMP IF X ZERO. Jump occurs if the X register is zero.

3.5.4 Control Conditional Jump Instructions

- JOR JUMP IF OVERFLOW RESET. Jump occurs if the OV equals zero.
- JOS JUMP IF OVERFLOW SET. Jump occurs if OV equals one. OV is reset to zero during jump.
- JSR JUMP IF SENSE INDICATOR RESET. Jump occurs if the SENSE indicator is off.
- JSS JUMP IF SENSE INDICATOR SET. Jump occurs if the SENSE indicator is on.

3.6 SHIFT INSTRUCTIONS

3.6.1 Instruction Format

Shift instructions move bit patterns in the computer registers either right or left. Shifts may involve a single register (A or X), a single register and the overflow (OV) register, or both the A and X registers and the OV indicator. The Processor provides logical, arithmetic and rotate shifts. Single-register shifts have a shift range of from 1 to 8 places while double-register shifts have a range of from 1 to 16 places. The single register shift instruction format is shown in Figure 3-11 and the instruction format for double register (long) shifts is shown in Figure 3-12.



Figure 3-11. Single Register Shift Format

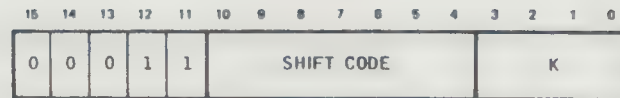


Figure 3-12. Double Register (Long) Shift Format

3.6.2 Arithmetic Shift Instructions

The shift paths for the arithmetic shift instructions are illustrated below in Figure 3-13 and Figure 3-14.



Figure 3-13. Arithmetic Right Shift

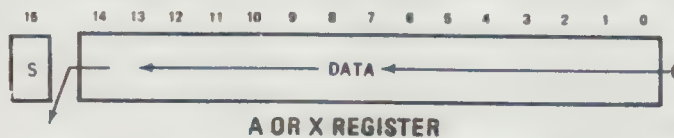


Figure 3-14. Arithmetic Left Shift

- ALA** ARITHMETIC SHIFT A LEFT. Contents of A Register (bits 0-14) are shifted left n places. The sign bit (bit 15) is unchanged. Zeros are shifted into bit 0 and bits shifted out of bit 14 are lost.
- ALX** ARITHMETIC SHIFT X LEFT. Contents of X Register (bits 0-14) are shifted left n places. The sign bit (bit 15) is unchanged. Zeros are shifted into bit 0 and bits shifted out of bit 14 are lost.
- ARA** ARITHMETIC SHIFT A RIGHT. Contents of A Register are shifted right n places. The sign bit (bit 15) is unchanged and is shifted into and propagated through bit 14. Bits shifted out of bit 0 are lost.



ARX ARITHMETIC SHIFT X RIGHT. Contents of X Register are shifted right *n* places. The sign bit (bit 15) is unchanged and is shifted into and propagated through bit 14. Bits shifted out of bit 0 are lost.

3.6.3 Logical Shift Instructions

The shift paths for the logical shift instructions are illustrated below in Figure 3-15 and Figure 3-16.

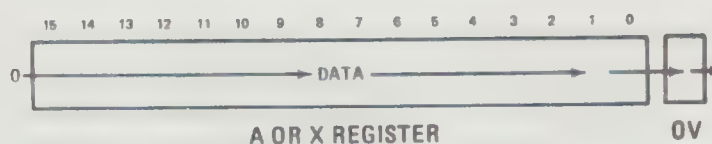


Figure 3-15. Rotate Right

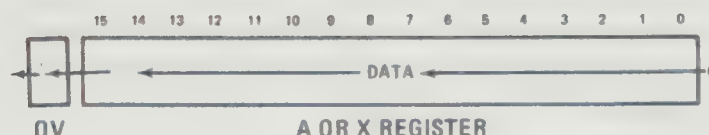


Figure 3-16. Rotate Left

- LLA** LOGICAL SHIFT A LEFT. Contents of A Register are shifted left *n* places through OV. Zeros are shifted into bit 0. Bits are shifted from bit 15 of A into OV. Bits shifted out of OV are lost. A and OV act as a 17-bit register.
- LLX** LOGICAL SHIFT X LEFT. Contents of X Register are shifted left *n* places through OV. Zeros are shifted into bit 0. Bits are shifted from bit 15 of X into OV. Bits shifted out of OV are lost. X and OV act as a 17-bit register.
- LRA** LOGICAL SHIFT A RIGHT. Contents of A Register are shifted right *n* places through OV. Zeros are shifted into bit 15. Bits are shifted from bit 0 of A into OV. Bits shifted out of OV are lost. A and OV act as a 17-bit register.
- LRX** LOGICAL SHIFT X RIGHT. Contents of X Register are shifted right *n* places through OV. Zeros are shifted into bit 15. Bits are shifted from bit 0 of A into OV. Bits shifted out of OV are lost. X and OV act as a 17-bit register.

3.6.4 Rotate Shift Instructions

The shift paths for the rotate shift instructions are illustrated below in Figure 3-17 and Figure 3-18.

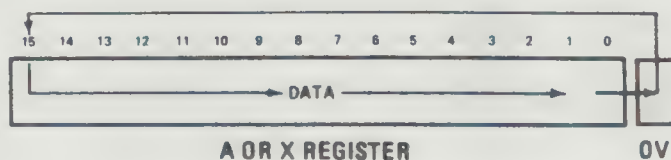


Figure 3-17. Rotate Right

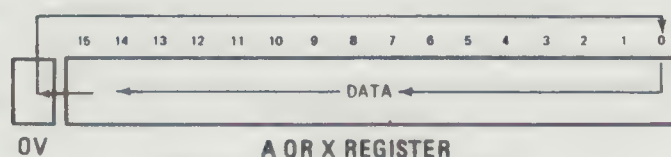


Figure 3-18. Rotate Left

- RLA ROTATE A LEFT WITH OVERFLOW. Contents of A Register are shifted left *n* places through OV. OV is shifted into bit 0 and bit 15 is shifted into OV. No bits are lost when this shift is executed. A and OV act as a 17-bit register.
- RLX ROTATE X LEFT WITH OVERFLOW. Contents of X Register are shifted left *n* places through OV. OV is shifted into bit 0 and bit 15 is shifted into OV. No bits are lost when this shift is executed. X and OV act as a 17-bit register.
- RRA ROTATE A RIGHT WITH OVERFLOW. Contents of A Register are shifted right *n* places through OV. OV is shifted into bit 15 and bit 0 is shifted into OV. No bits are lost when this shift is executed. A and OV act as a 17-bit register.
- RRX ROTATE X RIGHT WITH OVERFLOW. Contents of X Register are shifted right *n* places through OV. OV is shifted into bit 15 and bit 0 is shifted into OV. No bits are lost when this shift is executed. X and OV act as a 17-bit register.



3.6.5 Double Register (Long) Shift Instructions

The shift paths for the Long Logical Shift instructions are shown below in Figures 3-19 and 3-20.



Figure 3-19. Long Left Shift

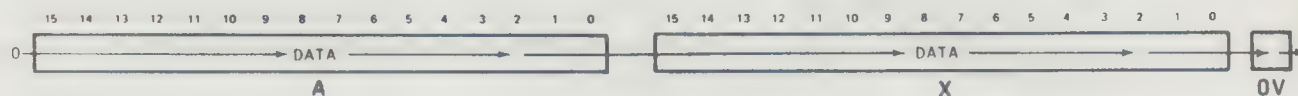


Figure 3-20. Long Right Shift

LLL LONG LOGICAL SHIFT LEFT. Contents of A and X Registers are logically shifted left through OV n places. For each bit position shifted, zero is shifted into X, X is shifted into A and A is shifted into OV. The previous contents of OV are lost. A, X and OV act as a 33-bit register.

LLR LONG LOGICAL SHIFT RIGHT. Contents of A and X Registers are logically shifted right through OV n places. For each bit position shifted, zero is shifted in A, A is shifted into X and X is shifted into OV. The previous contents of OV are lost. A, X and OV act as a 33-bit register.

3.6.6 Double Register (Long) Rotate Shift Instructions

Shift paths for the Long Rotate Shift instructions are shown below in Figures 3-21 and 3-22.

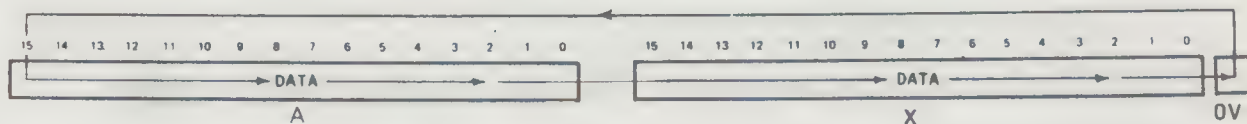


Figure 3-21. Long Rotate Right

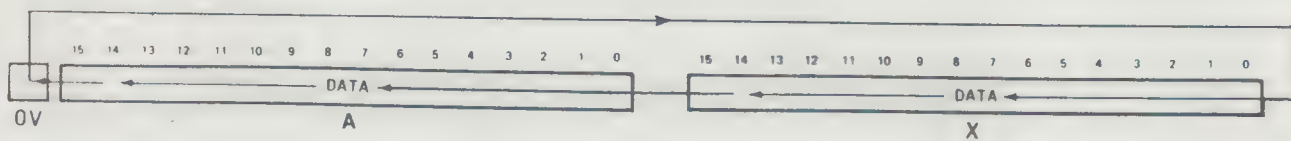


Figure 3-22. Long Rotate Left

LRL LONG ROTATE LEFT. Contents of A and X Registers are shifted left through OV *n* places. OV is shifted into X, X is shifted into A and A is shifted into OV. No bits are lost when this shift is executed. A, X and OV act as a 33-bit register.

LRR LONG ROTATE RIGHT. Contents of A and X Registers are shifted right through OV *n* places. OV is shifted into A, A is shifted into X and X is shifted into OV. No bits are lost when this shift is executed. A, X and OV act as a 33-bit register.

3.7 REGISTER CHANGE INSTRUCTIONS

3.7.1 Format

Register change instructions perform arithmetic and logical operations involving the A register, the X register and/or the overflow (OV) indicator. The Register Change Instruction format is shown in Figure 3-23.

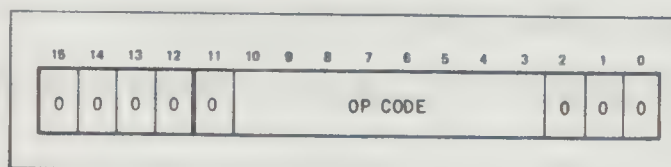


Figure 3-23. Register Change Format

3.7.2 A Register Change Instructions

ARM A REGISTER TO MINUS ONE. Sets the A register to -1 (:FFFF).

ARP A REGISTER TO PLUS ONE. Sets the A register to +1.



- CAR COMPLEMENT A REGISTER. Performs one's complement of the A register.
- DAR DECREMENT A REGISTER. Subtracts one from the A register. OV is set if arithmetic overflow occurs.
- IAR INCREMENT A REGISTER. Adds one to the A register. OV is set if arithmetic overflow occurs.
- NAR NEGATE A REGISTER. Performs two's complement of the A register. OV is set if arithmetic overflow occurs.
- ZAR ZERO A REGISTER. Sets the A register to zero.

3.7.3 X Register Change Instructions

- CXR COMPLEMENT X REGISTER. Performs one's complement of the X register.
- DXR DECREMENT X REGISTER. Subtracts one from the X register. OV is set if arithmetic overflow occurs.
- IXR INCREMENT X REGISTER. Adds one to the X register. OV is set if arithmetic overflow occurs.
- NXR NEGATE X REGISTER. Performs two's complement of the X register. OV is set if arithmetic overflow occurs.
- XRM X REGISTER TO MINUS ONE. Sets the X register to -1 (:FFFF).
- XRP X REGISTER TO PLUS ONE. Sets the X register to +1.
- ZXR ZERO X REGISTER. Sets the X register to zero.

3.7.4 OV Register Change Instructions

- COV COMPLEMENT OVERFLOW. Complements OV.
- LAO LSB OF A TO OVERFLOW. Bit 0 of the A register is copied into OV. The A register is unchanged.
- LXO LSB OF X TO OVERFLOW. Bit 0 of the X register is copied into OV. The X register is unchanged.



- ROV RESET OVERFLOW. Resets OV (to zero).
- SAO SIGN OF A TO OVERFLOW. Bit 15 of the A register is copied into OV. The A register is unchanged.
- SOV SET OVERFLOW. Sets OV (to one).
- SXO SIGN OF X TO OVERFLOW. Bit 15 of the X register is copied into OV. The X register is unchanged.

3.7.5 Multi-Register Change Instructions

- ANA AND OF A AND X TO A. The A and X registers are logically ANDed; result is placed in A. The X register is unchanged.
- ANX AND OF A AND X TO X. The A and X registers are logically ANDed; result is placed in X. The A register is unchanged.
- AXM A AND X REGISTERS TO MINUS ONE. Sets the A and X registers to -1 (:FFFF).
- AXP A AND X REGISTERS TO PLUS ONE. Sets the A and X registers to +1.
- CAX COMPLEMENT OF A TO X. Performs one's complement of A register and places result in the X register. The A register is unchanged.
- CXA COMPLEMENT OF X TO A. Performs one's complement of the X register and places result in the A register. The X register is unchanged.
- DAX DECREMENT A TO X. Subtracts one from the A register and places the result in X. The A register is unchanged. OV is set if arithmetic overflow occurs.
- DXA DECREMENT X TO A. Subtracts one from the X register and places the result in A. The X register is unchanged. OV is set if arithmetic overflow occurs.
- IAX INCREMENT A TO X. Adds one to the A register and places the result in X. The A register is unchanged. OV is set if arithmetic overflow occurs.
- IXA INCREMENT X TO A. Adds one to the X register and places the result in A. The X register is unchanged. OV is set if arithmetic overflow occurs.



- NAX NEGATE A TO X. Performs two's complement of the A register and places the result in X. The A register is unchanged. OV is set if arithmetic overflow occurs.
- NRA NOR OF A AND X TO A. Performs logical NOR of the A and X registers and places the result in A. The X register is unchanged.
- NRX NOR OF A AND X TO X. Performs logical NOR of the A and X registers and places the result in X. The A register is unchanged.
- NXA NEGATE X TO A. Performs two's complement of the X register and places the result in A. The X register is unchanged. OV is set if arithmetic overflow occurs.
- TAX TRANSFER A TO X. Transfers the A register to the X register. The A register is unchanged.
- TXA TRANSFER X TO A. Transfers the X register to the A register. The X register is unchanged.
- ZAX ZERO A AND X. Sets the A and X registers to zero.

3.7.6 Console Register Instructions

- ICA INPUT CONSOLE DATA REGISTER TO A. Loads the 16-bit contents of the Console Data register into the A register.
- ICX INPUT CONSOLE DATA REGISTER TO X. Loads the 16-bit contents of the Console Data register into the X register.
- ISA INPUT CONSOLE SENSE REGISTER TO A. Loads the 4-bit contents of the Console Sense register into the least significant 4 bits of the A register. The most significant 12 bits of the A register are set to zero.
- ISX INPUT CONSOLE SENSE REGISTER TO X. Loads the 4-bit contents of the Console Sense register into the least significant 4 bits of the X register. The most significant 12 bits of the X register are set to zero.
- OCA OUTPUT A TO CONSOLE DATA REGISTER. Transfers the A register to the 16-bit Console Data register. The A register is unchanged.
- OCX OUTPUT X TO CONSOLE DATA REGISTER. Transfers the X register to the 16-bit Console Data register. The X register is unchanged.



3.8 CONTROL INSTRUCTIONS

3.8.1 Format

Control instructions are used for general status manipulation in the computer. The general format for these instructions is shown in Figure 3-24.

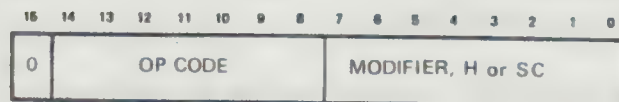


Figure 3-24. Control Format

3.8.2 Processor Control Instructions

HLT HALT. Halts the computer

NOP NO OPERATION. Causes a pause in the program

3.8.3 Mode Control Instructions

SBM SET BYTE MODE. Conditions the computer to address byte (8 bit) operands rather than word operands when executing Memory Reference instruction (see paragraph 3.2.2).

SWM SET WORD MODE. Conditions the computer to address word (16 bit) operands rather than byte operands when executing Memory Reference instructions (see paragraph 3.2.1). The "reset" condition of the computer is the Word Mode.

3.8.4 Status Control Instructions

The format of the 8-bit Computer Status Word is shown in Figure 3-25.

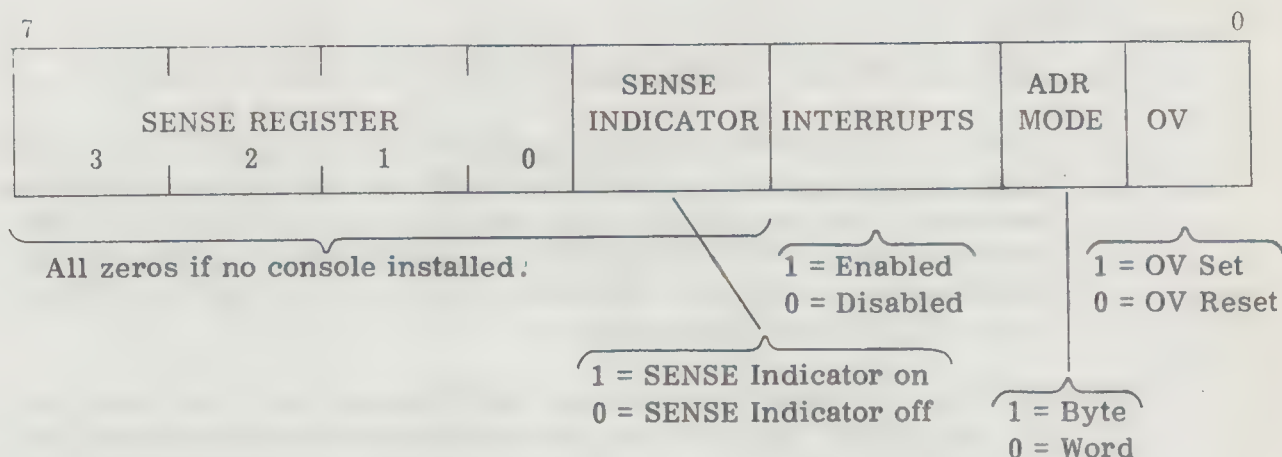


Figure 3-25. Computer Status Word Format

- SIA** STATUS INPUT TO A. Reads the Computer Status Word into the least significant 8 bits of the A register. Resets OV and sets the Address Mode to the Word Mode. The state of interrupts is unchanged. The most significant 8 bits of the A register are set to zero.
- SIN** STATUS INHIBIT. Inhibits interrupts and places the computer in the Word Mode for the number of succeeding instructions specified by the ID field. The ID field (bits 0-2) may be any value in the range 2 through 7. As an example, execution of the instruction SIN 5 will force Word Mode operation for the four succeeding instructions and will inhibit interrupt acknowledgement until completion of five succeeding instructions since interrupts are serviced at the end of instruction execution.
- SIX** STATUS INPUT TO X. Reads the Computer Status Word into the least significant 8 bits of the X register. Resets OV and sets the Address Mode to the Word Mode. The state of interrupts is unchanged. The most significant 8 bits of the X register are set to zero.
- SOA** STATUS OUTPUT FROM A. Writes the least significant 8 bits of the A register into the computer status register. However, this instruction does not alter the interrupt indicator.
- SOX** STATUS OUTPUT FROM X. Writes the least significant 8 bits of the X register into the computer status register. However, this instruction does not alter the interrupt indicator.



3.8.5 Interrupt Control Instructions

- CID** CONSOLE INTERRUPT DISABLE. Disables the Console interrupt.
- CIE** CONSOLE INTERRUPT ENABLE. Enables the Console interrupt. Console interrupts are generated each time the INT switch is pressed when the computer is in the Run mode. Console interrupts are also under the control of the EIN/DIN instructions. A special jumper option on the processor option card allows the console interrupt to be enabled independent of the EIN/DIN instructions.
- DIN** DISABLE INTERRUPTS. Prevents the Processor from responding to any interrupts. A special jumper option on the processor option card allows Power Fail, Console and Trap interrupt operation independent of DIN.
- EIN** ENABLE INTERRUPTS. Enables the recognition of external interrupts by the computer. This instruction does not take effect until completion of the next instruction in sequence.
- PFD** POWER FAIL INTERRUPT DISABLE. When the option placing Power Fail Interrupts outside EIN and DIN control is selected, the Power Fail Interrupt Disable (PFD) instruction inhibits recognition of Power Fail interrupts.
- PFE** POWER FAIL INTERRUPT ENABLE. When the option placing Power Fail Interrupt outside EIN and DIN control is selected, the Power Fail Interrupt Enable (PFE) instruction allows recognition of Power Fail interrupts. If Power Fail interrupts were disabled at the issuance of PFE, the PFE does not take effect until after two succeeding instructions have been executed.
- TRP** TRAP. Generates an interrupt to the Console interrupt location if interrupts are enabled or if the special jumper option placing Power Fail, Console and Trap interrupts outside EIN/DIN control is in use. In the latter case, there is no enable or disable instruction associated with the trap interrupt.

3.9 INPUT/OUTPUT INSTRUCTIONS

Input/Output instructions are either single word or multiple word instructions. All single word instructions use the same format, illustrated in Figure 3-26. Multiple word formats are described separately in Sections 3.9.4 and 3.9.5. All Input/Output instructions have 8 bits available for addressing a particular peripheral device and a particular register or function within a device. These 8 bits are arbitrarily divided



into a 5 bit Device Address field to address one of 32 devices and a 3 bit Function Code field to specify one of 8 registers or functions within a device.



Field	Bits	Description	Field	Bits	Description
Class	14, 15	These bits define the I/O instruction class.	Device Address	3-7	Used to select the specific peripheral device which is to respond to the I/O instruction.
Operation Code	8-13	These bits identify the specific I/O instruction that is being executed.	Function Code	0-2	Specified the function which the peripheral device is to perform. May also be used to identify a data source within the interface, or a status that is being sensed.

Figure 3-26. Single Word Input/Output Instruction Format

Both Word and Byte Input/Output instructions are available. Whether a full 16-bit word or an 8-bit byte is transferred depends upon the instruction used and is not effected by the word/byte addressing mode flip-flop (SWM/SBM) used by Memory Reference Instructions.

3.9.1 Control Input/Output Instructions

The Control Input/Output instructions are divided into Sense and Select instructions. Sense instructions are used to test the status of a function within the addressed peripheral device. Select instructions are used to control the operation of specific functions within the addressed peripheral device. The functions tested or controlled depend upon the individual peripheral device. Control Input/Output instructions use the Single Word Input/Output format as shown in Figure 3-26.

3.9.1.1 Sense Instructions

SEN SENSE AND SKIP ON RESPONSE. Tests the specified function in the specified peripheral device. If a true response is obtained, a one-word skip is generated. If a false response is obtained, the next instruction in sequence is executed.



SSN SENSE AND SKIP ON NO RESPONSE. Tests the specified function in the specified peripheral device. If a false response is obtained, a one-word skip is generated. If a true response is obtained, the next instruction in sequence is executed.

3.9.1.2 Select Instructions

SEL SELECT FUNCTION. Transmits the specified function code to the specified peripheral device along with a Select Control signal. All zeros are placed on the Data bus. Any action generated is a function of the interface design of the peripheral device.

SEA SELECT AND PRESENT A. Transmits the specified function code to the specified peripheral device along with a Select Control signal. The contents of the A register are placed on the Data bus. Any action generated is a function of the interface design of the peripheral device.

SEX SELECT AND PRESENT X. Transmits the specified function code to the specified peripheral device along with a Select Control signal. The contents of the X register are placed on the Data bus. Any action generated is a function of the interface design of the peripheral device.

3.9.2 Word Input/Output Instructions

Word Input/Output instructions transmit 16 bits of data at a time. They are divided into Unconditional and Conditional instructions. Conditional instructions are automatically repeated until a true sense response is obtained, at which time the data transmission occurs and the next instruction in sequence is executed. Response to an interrupt may occur "within" a conditional input/output instruction - i.e., during a false sense response an interrupt can be acknowledged and the computer will return to execution of the conditional input/output instruction after servicing the interrupt. If a word input is requested from an 8-bit device, the upper 8 bits will be input as zeros. If an output is performed to an 8-bit device the upper 8 bits will be ignored by the device.

3.9.2.1 Unconditional Word Input/Output Instructions

INA INPUT TO A REGISTER. Unconditionally transfers a full 16-bit data word from the addressed peripheral device to the A register.



- INAM** INPUT TO A REGISTER MASKED. Unconditionally transfers a full 16-bit data word from the addressed peripheral device to the A register. The data word is logically ANDed with the previous contents of the A register. The results are placed in the A register.
- INX** INPUT TO X REGISTER. Unconditionally transfers a full 16-bit data word from the addressed peripheral device to the X register.
- INXM** INPUT TO X REGISTER MASKED. Unconditionally transfers a full 16-bit data word from the addressed peripheral device to the X register. The data word is logically ANDed with the previous contents of the X register. The results are placed in the X register.
- OTA** OUTPUT A REGISTER. Unconditionally transfers the full 16-bit contents of the A register to the addressed peripheral device.
- OTX** OUTPUT X REGISTER. Unconditionally transfers the full 16-bit contents of the X register to the addressed peripheral device.
- OTZ** OUTPUT ZERO. Unconditionally transfers a 16-bit zero word to the addressed peripheral device.

3.9.2.2 Conditional Word Input/Output Instructions

- RDA** READ WORD TO A REGISTER. Senses the specified data source in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, a full 16-bit data word is transferred to the A register.
- RDAM** READ WORD TO A REGISTER MASKED. Senses the specified data source in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, a full 16-bit data word is transferred to the A register and logically ANDed with the previous contents of the A register. The results are placed in the A register.
- RDX** READ WORD TO X REGISTER. Senses the specified data source in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, a full 16-bit data word is transferred to the X register.



- RDXM** READ WORD TO X REGISTER MASKED. Senses the specified data source in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, a full 16-bit word is transferred to the X register and logically ANDed with the previous contents of the X register. The results are placed in the X register.
- WRA** WRITE FROM A REGISTER. Senses the specified condition in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, the full 16-bit contents of the A register are transferred to the addressed peripheral device.
- WRX** WRITE FROM X REGISTER. Senses the specified condition in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, the full 16-bit contents of the X register are transferred to the addressed peripheral device.
- WRZ** WRITE ZERO. Senses the specified condition in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, a 16-bit zero word is transferred to the addressed peripheral device.

3.9.3 Byte Input Instructions

Byte Input instructions input 8 bits of data to the least significant byte of a target register leaving the MSB byte unchanged. They are divided into Unconditional and Conditional instructions. Conditional instructions are automatically repeated until true sense responses are obtained, at which time the data transmission occurs and the next instruction in sequence is executed. Response to an interrupt may occur "within" a conditional input/output instruction - i.e., during a false sense response an interrupt can be acknowledged and the computer will return to execution of the conditional instruction after servicing the interrupt. Byte Input instructions use the Single Word Input/Output Instruction format as shown in Figure 3-26.

3.9.3.1 Unconditional Byte Input Instructions

- IBA** INPUT BYTE TO A REGISTER. Unconditionally transfers an 8-bit byte from the addressed peripheral device to the least significant 8 bits of the A register. The most significant 8 bits of the A register are unchanged.



IBAM INPUT BYTE TO A REGISTER MASKED. Unconditionally transfers an 8-bit byte from the addressed peripheral device to the least significant 8 bits of the A register. The data byte is logically ANDed with the previous contents of the least significant 8 bits of the A register. The results are placed in the least significant 8 bits of the A register and the most significant 8 bits of the A register are unchanged.

IBX INPUT BYTE TO X REGISTER. Unconditionally transfers an 8-bit byte from the addressed peripheral device to the least significant 8 bits of the X register. The most significant 8 bits of the X register are unchanged.

IBXM INPUT BYTE TO X REGISTER MASKED. Unconditionally transfers an 8-bit byte from the addressed peripheral device to the least significant 8 bits of the X register. The data byte is logically ANDed with the previous contents of the least significant 8 bits of the X register. The results are placed in the least significant 8 bits of the X register and the most significant 8 bits of the X register are unchanged.

3.9.3.2 Conditional Byte Input Instructions

RBA READ BYTE TO A REGISTER. Senses the specified data source in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, an 8-bit data byte is transferred to the least significant 8 bits of the A register. The most significant 8 bits of the A register are unchanged.

RBAM READ BYTE TO A REGISTER MASKED. Senses the specified data source in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, an 8-bit data byte is transferred to the least significant 8 bits of the A register and logically ANDed with the previous contents of the least significant 8 bits of the A register. The results are placed in the least significant 8 bits of the A register and the most significant 8 bits of the A register are unchanged.

RBX READ BYTE TO X REGISTER. Senses the specified data source in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, an 8-bit data byte is transferred to the least significant 8 bits of the X register. The most significant 8 bits of the X register are unchanged.



RBXM READ BYTE TO X REGISTER MASKED. Senses the specified data source in the addressed peripheral device. If a false response is received, the instruction is repeated (and interrupts may be acknowledged). When a true response is received, an 8-bit data byte is transferred to the least significant 8 bits of the X register and logically ANDed with the previous contents of the least significant 8 bits of the X register. The results are placed in the least significant 8 bits of the X register and the most significant 8 bits of the X register are unchanged.

3.9.4 Block Input/Output Instructions

The two instructions in this class provide for high-speed, full 16-bit word data transfers. The Processor is totally dedicated to these instructions until the specified block of data has been completely transferred - i.e., no interrupts may be serviced until the instructions have been executed to completion.

The Block Transfer instructions are double-word instructions. The second word of the instruction contains the base address minus one of the associated data buffer in memory. The X register contains the (positive) number of words to be transferred - i.e., the length of the data buffer. The memory location of each word transferred is obtained by summing the base address minus one and the contents of the X register. As each data word is transmitted, the X register is decremented by one. Thus, the data buffer is output or input in descending order, beginning with the highest memory location and ending with the lowest memory location (base address plus length -1). When the X register is decremented to zero, the next instruction in sequence is executed.

The format for the Block Transfer instructions is shown in Figure 3-27.

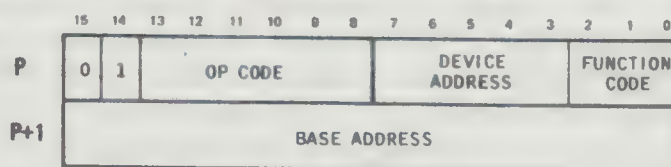


Figure 3-27. Block Input/Output Instruction Format

BIN BLOCK IN. Senses the specified data source in the addressed peripheral device and inputs a full 16-bit data word from the selected device each time a true sense response is received. The instruction executes until all data words have been input. Interrupts may be acknowledged only after completion of the instruction. The A register is unchanged.



BOT BLOCK OUT. Senses the specified data source in the addressed peripheral device and outputs a full 16-bit data word to the selected device each time a true sense response is received. The instruction executes until all data words have been output. Interrupts may be acknowledged only after completion of the instruction. The A register is unchanged.

3.9.5 Automatic Input/Output Instructions

The Automatic Input/Output instructions (Auto I/O) provide data transfers directly between memory and peripheral devices without affecting the A and X registers. These multiple word instructions effectively constitute complete I/O subroutines, thus facilitating their use as interrupt instructions. They increment a (negative) data word or byte counter, increment a data word or byte pointer and transfer a data word or byte between memory and a peripheral device.

Each Auto I/O instruction occupies three words in memory. The first word contains the instruction itself, the second word contains the two's complement (negative) of the word or byte count for the data buffer and the third word contains an address pointer specifying the location one less than the first (lower-order memory) location of the data buffer. The data buffer is input or output in order of ascending memory locations (low-order to high-order). The format for these instructions is shown in Figure 3-28.

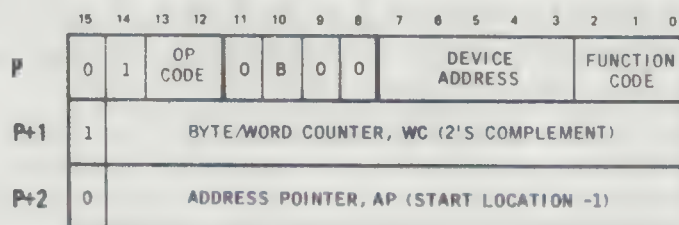


Figure 3-28. Automatic Input/Output Instruction Format

The second word represents the negative (two's complement) data word or byte count for the buffer being transmitted. This word is incremented once prior to each data word or byte transfer and must be preset each time a block of data is to be transferred.

The third word of the instruction is an address pointer specifying the byte or word location one less than the start of the data buffer. This word is incremented once prior to each data word or byte transferred and must be preset each time a block of data is to be transferred.



Operation of Automatic I/O Instructions differ depending upon usage. When used as a normal in-line program instruction, the Automatic I/O instruction sequence is as shown in Figure 3-29. Each time the instruction is executed the word/byte count and address pointer are incremented, one word or byte of data is transferred and then the incremented word count is examined. If the word count has not yet reached zero, the next instruction executed is from location P+4. If the word count reached zero, the next instruction executed is at location P+3 (End of Block exit location). Since Automatic I/O instructions do not sense for the peripheral device to be ready prior to data transfer, a sense instruction should be used prior to each execution (one word transferred) of the instruction, i.e., to transfer a block location, P+4 would normally contain a jump back to a sense instruction prior to location P.

P	Automatic I/O Instruction
P+1	Word/Byte Counter (negative)
P+2	Address Pointer (starts address -1)
P+3	End of Block Exit (Word Count = 0)
P+4	Next Instruction (Word Count $\neq$ 0)

Figure 3-29. In-line Auto I/O Instruction Sequence

Automatic I/O instructions may also be used under interrupt control at an interrupt location to implement a Direct Memory Channel. In this application, the Automatic I/O instruction is executed once each time the peripheral device indicates that it is ready for a data transfer by interrupting to the location containing the Automatic I/O instruction. Since the Automatic I/O instructions do not alter any processor registers, no jumping to an interrupt subroutine to save registers, status, and return location is required. The Automatic I/O instruction is, itself, a one word subroutine. When executed under interrupts, the skips after execution are suppressed. Instead, if the word count has not reached zero after a data transfer, control is passed directly back to the main-line program at the point it was interrupted. If the word count did reach zero, a special signal (ECHO-) is sent to the peripheral device to indicate that it should stop requesting further data transfers. The Automatic I/O instruction transfers control back to the main-line program whether ECHO is set or not. Upon receipt of ECHO, the peripheral device stops data transfer requests, performs any stop action required (i.e. CRC checking or generation for magnetic tape), and then generates an End-of-Block interrupt so that the program can process the data block input or prepare another block for output. Although the End-of-Block interrupt can be vectored to any location by the peripheral controller, it is standard practice for the controller to vector this interrupt to four locations beyond the data transfer interrupt location. Figure 3-30 illustrates the typical usage of Automatic I/O instructions under interrupts.



Data Transfer Interrupt Location	I	Automatic I/O Instruction
	I+1	Word/Byte Counter (negative)
	I+2	Word/Byte Counter (negative)
	I+3	Unused
	I+4	JST* $\$$ -1 (jump and store to END-of-Block subroutine)

Figure 3-30. Interrupt Location Auto I/O Instruction Sequence

- AIB** AUTOMATIC INPUT BYTE TO MEMORY. Increments the byte counter and the byte address pointer and unconditionally inputs one 8-bit byte from the specified data source in the addressed peripheral device to the updated byte location in memory addressed by the address pointer. When the byte count is incremented to zero, the normal one-word skip after the data transfer does not take place, or when used as an interrupt instruction an Echo signal to the addressed peripheral device is generated.
- AIN** AUTOMATIC INPUT WORD TO MEMORY. Increments the data word counter and the address pointer and unconditionally inputs a full 16-bit data word from the specified data source in the addressed peripheral device to the updated word location in memory addressed by the address pointer. When the word count is incremented to zero, the normal one-word skip after the data transfer does not take place, or when used as an interrupt instruction an ECHO signal to the addressed peripheral device is generated.
- AOB** AUTOMATIC OUTPUT BYTE FROM MEMORY. Increments the byte counter and the byte address pointer and unconditionally outputs one 8-bit byte from the updated byte location in memory addressed by the byte address pointer to the specified data source in the addressed peripheral device. When the byte count is incremented to zero, the normal one-word skip does not take place, or when used as an interrupt instruction, an ECHO signal to the addressed peripheral device is generated.
- AOT** AUTOMATIC OUTPUT WORD FROM MEMORY. Increments the data word counter and the address pointer and unconditionally outputs a full 16-bit data word from the updated word location in memory addressed by the address pointer to the specified data source in the addressed peripheral device. When the word count is incremented to zero, the normal one-word skip after the data transfer does not take place, or when used as an interrupt instruction, an Echo signal to the addressed peripheral device is generated.

1911

1911

1911

1911

1911

1911

1911

1911



Section 4

MEMORY

4.1 GENERAL

A family of memory modules may be used within the ALPHA LSI computer, ranging from core through MOS, high speed to low speed, large bulk storage to small high performance scratchpad memories, allowing the tailoring of the computer system to specific applications while optimizing the price/performance ratio.

The Memory/Processor interface is asynchronous, sometimes referred to as a "handshake" interface. As a result, Memory Modules of varying access and cycle times may be mixed in the same system. The computer performance is to a large degree a function of the memory performance.

4.2 BASIC MEMORY

The NAKED MINI 16 LSI computer in its standard configuration includes 4K x 16 of core memory.

System access time: 600 nanoseconds max.

System cycle time: 1.6 microseconds max.

Optionally, other memory configurations are available in the NAKED system: 1K x 16 MOS, 2K x 16 MOS, 8K x 16 Core, etc.

4.3 MEMORY EXPANSION

In addition to the memory included on the NAKED MINI 16 LSI computer module, other memory modules may be installed in a chassis to form the ALPHA 16 LSI computer system.

Memory modules of various technologies, performances, and storage sizes may be mixed in any configuration in the chassis, so long as the memory bank control chain is kept intact.

The standard ALPHA 16 LSI system may address up to 32K word of memory. The addition of an optional I/O controller to the system provides a memory banking capability, allowing memory expansion limited only by the number of chassis used to house memory modules.



4.4 INTERCHANGEABILITY

Within a computer system, memory modules may be reconfigured in the various slots with no adjustments.

Memory modules among various ALPHA 16 LSI computer systems may be interchanged with no adjustment, and as long as the memories interchanged are of the same type (technology, performance, storage size), system performance is unchanged.

4.5 MEMORY-TO-PROCESSOR INTERFACE

The memory subsystem interface to the Processor utilizes the I/O bus for data and address communication. The memory interface is described with respect to DMA operation in section 5. The memory-to-Processor interface is not specified in this document.

4.6 MEMORY SYSTEM ACCESS TIME

Memory system access time is different from raw memory access time due to Processor synchronization time. The Processor repeats 200 nanosecond phases after issuing the memory start signal until a memory acknowledge is received. The following table provides the relationship between raw memory access time (t_{ma}) and memory system access time (t_{sa}):

t_{ma}	t_{sa}
0-133 ns	200 ns
134-333	400
334-533	600
534-733	800
734-933	1000

When computing instruction execution rates, the appropriate memory system access time figure must be included for each memory access.



Section 5

INPUT/OUTPUT, MEMORY INTERFACE

5.1 GENERAL

The NAKED MINI 16 LSI computer provides a standard I/O and memory interface. The data and address busses are time shared for both memory and I/O operations. Separate control lines are provided for each subsystem. This external I/O bus is wired to all slots of the ALPHA 16 LSI motherboard, except the Power Supply slot.

5.2 EXTERNAL I/O BUS COMPONENTS (Figure 2-1)

The NAKED MINI 16 LSI Computer external I/O bus consists of three major components: the Address bus (A), the Data bus (D), and the Control bus (C).

5.2.1 Address Bus (A)

The Address bus consists of 16 lines (AB00- through AB15-) that are time shared by the Processor and DMA controllers.

The Processor and DMA controllers use the 15 bits of the A bus to address memory locations. The 16th A bus bit is used to specify word or byte memory operation. During I/O operations, the Processor uses the low order 8-bits of the A bus to convey device address and function code information to I/O devices. The high order 8-bits contain random information and are not normally used. The format of the low order 8-bits during I/O operations is as follows:

AB07-	Device Address bit 4	} P Field
AB06-	Device Address bit 3	
AB05-	Device Address bit 2	
AB04-	Device Address bit 1	
AB03-	Device Address bit 0	
AB02-	Function Code bit 2	} F Field
AB01-	Function Code bit 1	
AB00-	Function Code bit 0	

NOTE

The eight lines devoted to the Device Address and Function Code are arbitrarily divided into groups of five and three, respectively. They can be divided differently to increase or decrease the number of device addresses and function codes. For example, six lines can be devoted to the Device Address and only two to the Function Code. This would increase the number of device addresses to 64 and reduce the number of function codes to 4.

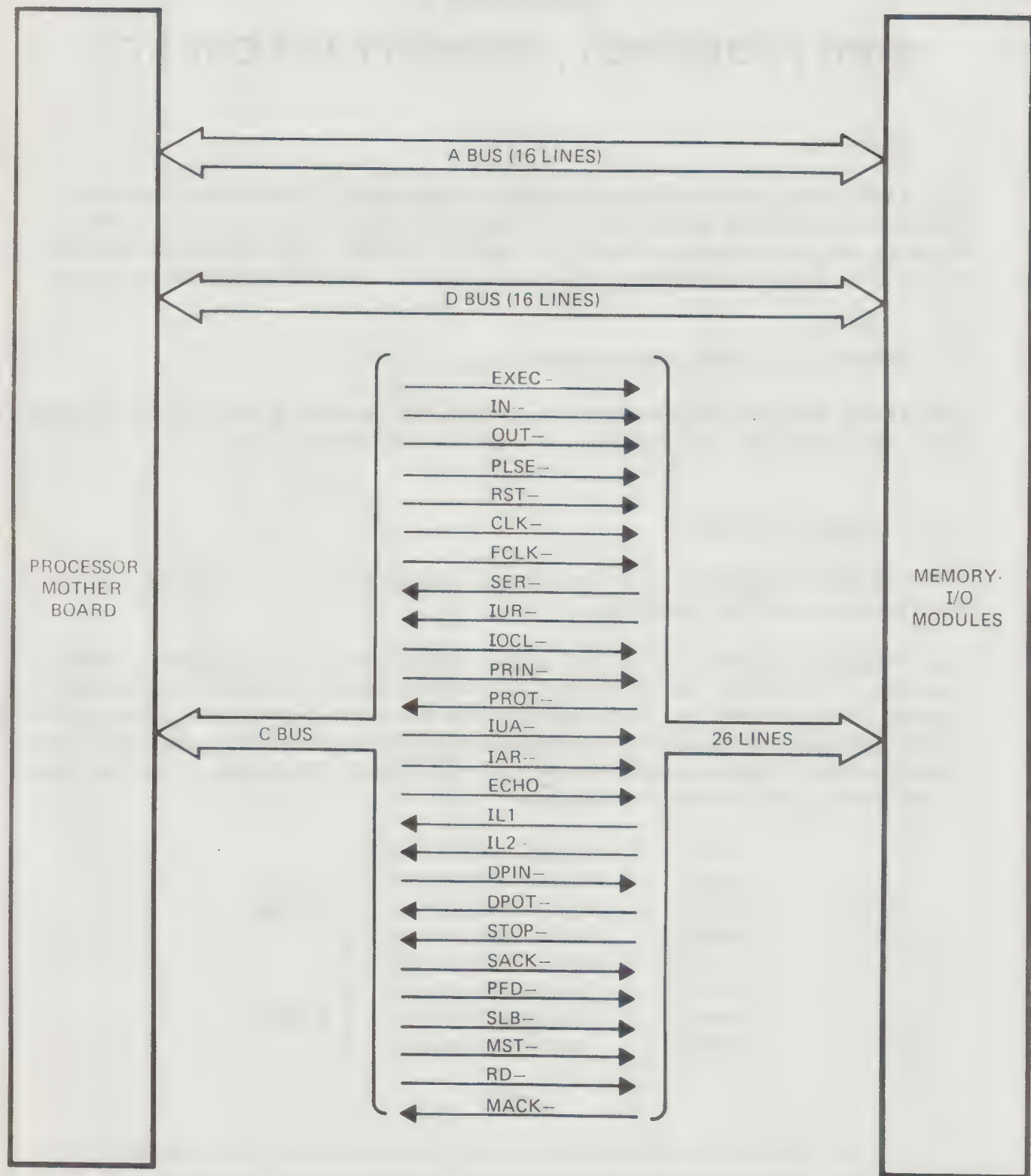


Figure 5-1. I/O Bus Components



Throughout the remainder of this design guide, all examples which involve I/O addresses assume the arbitrary five and three division.

5.2.2 Data Bus (D)

The Data bus consists of 16 bidirectional lines (DB00- through DB15-) that are time shared by the Processor, memory, and I/O interface controllers.

The Processor uses the D bus to read data from or write data into memory. Likewise, the Processor uses the D bus to transfer data to or from an I/O interface controller.

A DMA controller uses the D bus to read data from or write data into memory.

I/O interfaces use the D bus to convey an interrupt address to the Processor during interrupt processing.

5.2.3 Control Bus (C)

The C bus consists of 26 unidirectional control lines which define the specific action that an interface device is to perform. Eighteen lines are outputs from the Processor to all memories and interface controllers while eight lines are inputs from either memories or interface controllers to the Processor. The 26 C bus lines are subdivided into four broad categories: I/O command; utility signals; interrupt signals; and DMA signals. Except as noted below, all Processor generated or received signals may also be generated or received by DMA controllers during DMA operations.

5.2.3.1 I/O Commands

There are three signals in this category: EXEC-, IN- and OUT-. These signals define the type of I/O operation in process.

- | | |
|-------|---|
| EXEC- | Execute is a Processor generated signal that indicates that the current instruction is a Select or Select and Present instruction. EXEC- is used typically to set or reset controls in the addressed interface. |
| IN- | Input is a Processor generated signal that indicates that the current instruction is an input instruction and that the addressed interface should place input data on the data bus. |
| OUT- | Output is a Processor generated signal that indicates that the current instruction is an output instruction and that the Processor has placed output data on the data bus for the addressed interface controller to accept. |



5.2.3.2 Utility Signals

There are five signals in this category: PLSE-, RST-, CLK-, FCLK- and SER-.

- PLSE- Pulse is a Processor generated signal which is used as a strobe pulse to load registers during an output transfer, set or reset controls during a Select instruction, reset data transfer controls during an input transfer, and to reset Interrupt Stimulus Store controls upon recognition of an interrupt.
- RST- System Reset is a Processor or console generated signal which is used to reset all controls in ALL interfaces to a known starting configuration. RST- is generated by the Processor in response to a power failure condition, an Autoload Initiation sequence, or when the console RESET switch is depressed. Note - not driven by DMA controllers.
- CLK Clock is a Processor generated, 1 megahertz, free-running square wave signal that may be used as a timing reference by interface controllers--it is not synchronized to Processor operations. Note that only the Processor operates this signal. DMA controllers may not generate this signal.
- FCLK- Fast clock is a Processor generated, 10 megahertz, free running square wave signal that is used as a timing reference by DMA controllers--it is not synchronized to Processor operations.
- SER- Sense Response is a signal generated by an addressed interface controller to convey status to the Processor.

5.2.3.3 Interrupt Signals

There are nine signals associated with interrupt generation and processing. These signals are: IUR-, IOCL-, PRIN-, PROT-, IUA-, ECHO-, IL1-, and IL2-.

- IUR- Interrupt Request is a multiplexed interrupt request line which multiple interfaces generate to request interrupt service. All interfaces which use this line are forced to compete with each other for recognition by the Processor. If two or more interfaces request interrupt service at the same time, recognition is given to the highest priority interface via the priority string (PRIN- and PROT-).



- IOCL-** I/O Clock is a Processor generated signal which is used by interfaces to synchronize IUR interrupt requests into the Processor. IOCL has a minimum duration of 150 nanoseconds; however, the duration varies with internal Processor operation. When an interrupt is recognized by the Processor, IOCL is inhibited to prevent the generation of additional IUR interrupt requests. IOCL remains inhibited until the Processor completes execution of the interrupt instruction.
- PRIN- and PROT-** Priority In and Priority Out. PRIN- and PROT- form an interrupt priority chain which is strung serially through all interface controllers and memories. PRIN- is the name given to the priority chain where it enters an interface. If low, it allows the interface to generate interrupts. Each interface generates a PROT- signal to indicate that neither it nor other upstream devices are generating an interrupt. The PROT- signal from each interface is the PRIN- signal for the next downstream interface.
- IUA-** Interrupt Acknowledge is a Processor generated signal which goes true upon recognition of any interrupt and remains true during execution of the interrupt instruction.
- IAR-** Interrupt Address Request is a Processor generated signal which is used to request an interrupt address from an interface in response to an interrupt request.
- ECHO-** Echo is generated by the Processor when an Auto I/O instruction has transferred all data or by an IMS instruction when the count overflows. ECHO- is typically used by the interface to request an interrupt. This interrupt vectors to a user-determined location in memory which normally contains a JST instruction to a sub-routine which performs the necessary housekeeping associated with an end-of-block or elapsed count operation.
- IL1- and IL2-** Interrupt Lines 1 and 2 are interface generated high priority interrupt request lines which interrupt to locations : 0002 and : 0006, respectively. They are higher priority than the IUR line. IL1 has priority over IL2. IL1 and IL2 do not require interrupt vectoring by the interface as does IUR.

5.2.3.4 DMA Signals

Nine signals are associated with DMA control and processing. These signals are: DPIN-, DPOT-, STOP-, SACK-, PFD-, SLB-, MST-, RD- and MACK-.



5.2.3.4.1 DMA Controller-to-Processor Interface. Five control lines comprise the interface between the DMA controller and the Processor.

- DPIN- and DPOT- DMA Priority In and DMA Priority Out. DPIN and DPOT form a DMA priority chain which is strung serially through all DMA controllers and memories. DPIN- is the name given to the priority chain where it enters a DMA controller. If low, it allows the controller to access memory. Each controller generates a DPOT- signal to indicate that neither it nor other upstream controllers are communicating with memory. The DPOT- signal from each controller is the DPIN- signal for the next downstream controller. The DPIN- and DPOT- signals are strong through the "200" side of the motherboard only (see paragraph 2.4).
- STOP- Stop Processor is a DMA controller generated signal which stops the Processor upon completion of its current machine cycle to permit the DMA controller to gain control of the I/O bus. STOP- may be generated at any time and may remain active for any length of time.
- SACK- Stop Acknowledge is a negative-true Processor generated signal which informs DMA controllers that the Processor is no longer controlling the I/O bus and that the DMA controllers may take over the I/O bus. SACK- will remain true until STOP- is removed.
- PFD- Power Failure Detected is a power supply generated signal which when active forces any DMA operations to terminate in order to permit the Processor to shut down the system in an orderly manner.

5.2.3.4.2 DMA Controller-to-Memory Interface. The DMA controller may drive the interface between itself and the memory only when SACK- is true and DPIN- is true. The interface between the DMA controller and the memory is comprised of four control lines, 16 data bus lines, and 16 address bus lines.

- SLB- Select Least Significant Byte is a Processor or DMA controller generated signal which is used for Byte Mode memory accesses. When SLB- is low, the least significant byte (bits 0 through 7) of the addressed memory word is accessed. When SLB- is high, the most significant byte (bits 8 through 15) of the addressed memory word are accessed. SLB- is used to disable memory during Autoload operations by forcing it low while AB15- is high (word mode).
- MST- Memory Start is issued by a DMA controller to initiate a memory cycle. MST- must remain true until the leading edge of MACK-, and may be removed after the leading edge of MACK-.



Typically, MST- is issued no sooner than 200 nanoseconds after SACK- comes true to allow for priority-ripple time. Priority resolution also occurs after every MACK- to allow a higher priority DMA controller to bump a lower priority controller. To allow for this priority-ripple time, MST- may be issued no sooner than 200 nanoseconds after the trailing edge of MACK for memory cycles subsequent to the first cycle.

MACK- Memory Acknowledge is issued by the memory when:

1. data is available on the data bus during a read cycle, or
2. data, address and control information has been saved by the memory during write cycles.

In either case, MACK- will be held true until at least the trailing edge of MST- and no later than 75 nanoseconds after MST- is removed. When MST- is removed, the memory will also remove data from the data bus within 75 nanoseconds.

RD- Read Cycle is issued by the DMA controller no later than 25 nanoseconds after the leading edge of MST- to instruct the memory to perform a read cycle; otherwise, a write cycle will be performed. RD- must remain true for the read cycle until the leading edge of MACK-, and removed within 75 nanoseconds after the trailing edge of MST-.

5.3 I/O TRANSFER TIMING (Figure 5-2)

All timing specifications in the following paragraphs apply under full load conditions (all slots occupied) and maximum line length conditions (console connected and external bus expansion cable connected and terminated at the buffer card). All signals are referenced to slot A of the mainframe motherboard.

All references to the A bus are to the lower eight bits (AB00- through AB07-) only. The I/O transfer timing is shown in figure 5-2.

All I/O to Processor communication may be classified in the following groups:

1. Input cycle
2. Output cycle
3. Sense cycle
4. Control cycle
5. Interrupt cycle (interrupt address request)



5.3.1 Input Timing

Data must be valid on the data bus (DBXX-) no later than 300 nanoseconds after the leading edge of IN-.

For conditional inputs (e.g., RDA), the sense response (SER-) must be valid no later than 275 nanoseconds after the leading edge of the address. The conditional I/O instruction is continually repeated if SER- is not generated during the I/O cycle.

Data may be removed no sooner than the leading edge of PLSE- and no later than 75 nanoseconds after the trailing edge of IN-.

5.3.2 Output Timing

Data will be valid on the data bus at the same time as the leading edge of OUT-. For conditional outputs (e.g., WRA), the sense response (SER-) must be valid no later than 275 nanoseconds after the leading edge of the address. The conditional I/O instruction is continually repeated if SER- is not generated during the I/O cycle.

5.3.3 Sense Timing

The sense response (SER-) must be valid no later than 275 nanoseconds after the leading edge of the address.

Signal PLSE- is not issued for the sense instructions SEN and SSN.

5.3.4 Control Timing

For the select instructions (SEL, SEA, SEX), the data lines will be valid at the same time as the leading edge of EXEC-. For the SEL instruction, this data will be all zeros. The SEA and SEX instructions are similar to the output instructions OTA and OTX, respectively, except signal EXEC- is issued instead of OUT-.

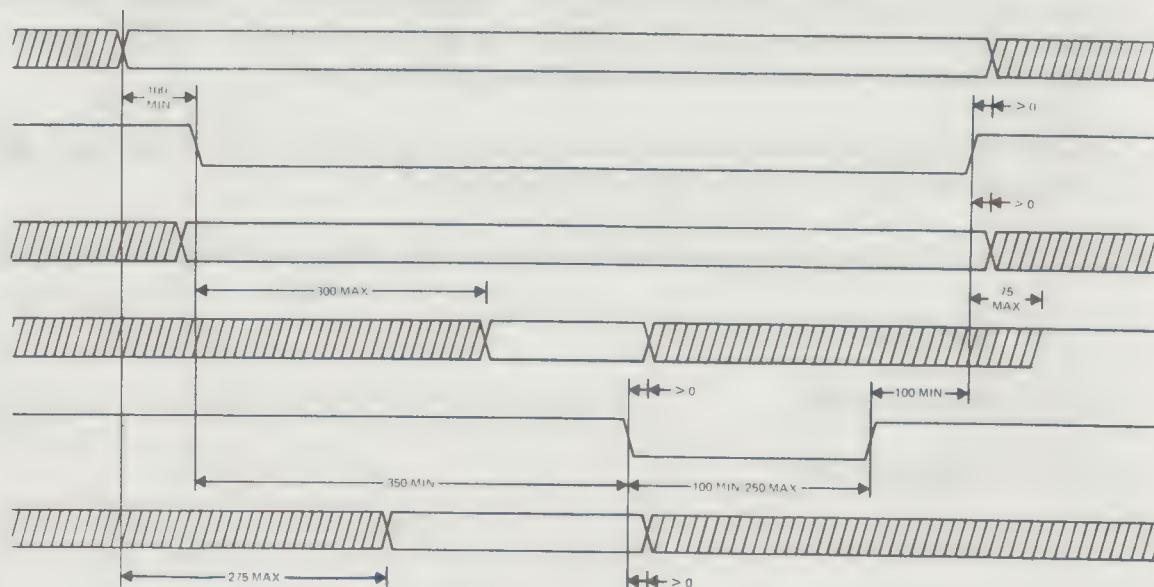


Figure 5-2. I/O Transfer Timing



5.4 INTERRUPTS

The ALPHA LSI computer uses a vectored interrupt structure whereby each interrupt has associated with it a memory address.

Four interrupts, PF (Power Fail), CI (Console), IL1 and IL2, are fully implemented within the Processor. In addition, a universal interrupt request line (IUR) is implemented in the system. The IL1, IL2 and IUR interrupts are provided for general usage in the peripheral controller subsystem. The power fail and console interrupts are special purpose interrupts which are not available for peripheral controller usage.

The IL1 and IL2 interrupts have dedicated memory addresses associated with them; these addresses may be modified by the interrupting controller. The IUR interrupt requires that the interrupting controller provide a memory address to the Processor during the interrupt address request cycle.

A computer instruction (TRAP) causes a console interrupt to take place, just as if an interrupt had been generated from the console.

The memory address associated with the interrupt is referred to as the interrupt address. From this location, the interrupt instruction is fetched during interrupt processing. Interrupt processing takes place when an interrupt is recognized but only after completion of the current instruction. Interrupt processing consists of interrupt recognition, an interrupt address request cycle, interrupt instruction fetch from memory and finally interrupt instruction execution. Unless the interrupt instruction itself modifies the program counter (P register), execution of the mainline program will continue with the next mainline instruction after the interrupt instruction has been completed, even if other interrupts are pending. At least one mainline instruction will be executed between interrupt instructions, regardless of the status of pending interrupts.

5.4.1 Interrupt Enabling and Disabling

Under program control (instructions EIN and DIN), interrupt recognition can be enabled or disabled. When in the disable state, all interrupts including the power fail interrupt will be ignored. When the computer is returned to the interrupt enable state, the highest priority interrupt pending at that time will be serviced.

An option is provided which allows the user to place the power fail, console and TRAP interrupts outside the control of the EIN/DIN instructions. The option is controlled by a special option-jumper connector which is installed on the rear edge of the NAKED MINI LSI option card. If this option is selected, then the TRAP interrupt will be unconditionally recognized as a console interrupt, and the power fail and true console interrupts are under the control of dedicated enable/disable instructions only:



PFE	Enable power fail interrupt recognition
PFD	Disable power fail interrupt recognition
CIE	Enable console interrupt recognition
CID	Disable console interrupt recognition

Note that the enable/disable aspect of the console and TRAP interrupts under this option is the only functional difference between these two kinds of interrupts: TRAP is unconditionally recognized if the option is selected, regardless of CIE/CID status. If the option is not selected, TRAP is controlled only by EIN and DIN instructions, while the Console Interrupt is controlled by EIN/DIN and CIE/CID instructions.

The EIN/DIN control may be set to the enable state via execution of the EIN instruction only. Similarly, the CIE/CID and PFE/PFD status is controlled in a manner similar to the EIN/DIN control, except the JST instruction has no effect on these controls.

5.4.2 Interrupt Masking

Interrupts may be masked via computer logic as a function of machine operation, independent of enable/disable controls.

Certain ALPHA LSI instructions may not be interrupted; that is, upon completion of the uninterruptable instruction, an interrupt will not be recognized and an interrupt instruction, therefore, will not be executed until after an interruptable instruction is executed.

DIN
EIN, if interrupts were previously disabled
HLT

Furthermore, interrupts will not be recognized when the computer has been stopped via the console or during direct memory access (DMA) operation.

5.4.3 Reserved Memory Locations

Certain memory locations are reserved for specific interrupts and should not in general be used by the main program. These locations are:

: 0000 Power fail restart. Used as the start location when power is returned. Power-up restart is not really an interrupt but instead is referred to as a "pseudo-interrupt".



: 0002	Interrupt Line 1 (IL1). An IL1 interrupt automatically vectors to this location unless the interrupting controller modifies the interrupt address during the interrupt address request cycle.
: 0006	Interrupt Line 2 (IL2). An IL2 interrupt automatically vectors to this location unless the interrupting controller modifies the interrupt address during the interrupt address request cycle.
: 001C	Power Fail interrupt (PF). A power fail interrupt automatically vectors to this location.
: 001E	Console and TRAP interrupt (CI). A console or TRAP interrupt automatically vectors to this location.
: XXXX	For IUR interrupts, the interrupt address is determined by the interrupting controller; no "standard" interrupt location is assigned.

Additional interrupt locations may be used by computer options or peripheral controllers. Refer to the ALPHA 16 LSI and NAKED MINI 16 LSI I/O Interface Design Guide for a complete listing of interrupt locations reserved for standard options.

5.4.4 Interrupt Address Offset

To provide flexibility in programming with respect to memory base page usage, an option is provided which allows the user to cause interrupt addresses to be offset for certain interrupts. The option is controlled by a special option-jumper connector which is installed on the rear edge of the NAKED MINI 16 LSI option card. If this option is selected, an offset of : 100 locations occurs for certain interrupts so that the following interrupt location vectoring occurs:

: 0102	IL1
: 0106	IL2
: 011C	PF
: 011E	CI

Note that IUR interrupts are unaffected by the interrupt offset option. (On the option card, the capability exists for offsetting interrupts generated via that card by : 100 locations, even though the IUR line is used. This offset option is independent of the primary offset described above. The only common aspect of the two offset options is that they share the same option-jumper connector for option selection.)



5.4.5 Interrupt Priority

The ALPHA LSI computer recognizes interrupts within the following priority structure (listed from highest priority to lowest priority):

1. Power Fail (PF)
2. Console (CI, TRAP)
3. Interrupt Line 1 (IL1)
4. Interrupt Line 2 (IL2)
5. Universal Interrupt (IUR)

The IUR interrupt line has associated with it a priority string (a daisy-chained priority signal through all motherboard connectors) for resolving interrupt priority where several IUR interrupts may occur simultaneously. The Processor turns off the priority signal during the interrupt address request cycle when processing interrupts of higher priority than IUR. Only the interrupting peripheral controller receiving a true priority signal may provide an interrupt address during the interrupt address request cycle. In this event, this peripheral controller must block the priority signal from "downstream" controllers to prevent priority conflict.

Because of the priority string, the peripheral controller closest to the Processor along the priority string receives the highest priority. In other words, each motherboard connector has associated with it an IUR priority level (listed highest priority first):

Expansion Connector

A100	expansion chassis only
A200	
B200	
B100	
C100	
C200	
D200	
D100	
E100	
E200	

Expansion Connector

Each I/O controller, regardless of interrupt requirements on that controller, must pass priority downstream to maintain the interrupt priority structure of the machine.

In peripheral controller design, it may become necessary to detect interrupt generation of higher priority with respect to IL1 or IL2; the interrupt address request cycle itself has associated with it no control lines for detecting higher priority interrupt generation. However, the peripheral controller can detect such a case by monitoring bit 4 of the data bus (DB04-) during the interrupt address request cycle. If a power fail, console or TRAP interrupt is pending, DB04- will be forced to ground during the interrupt address request cycle, since this address bit is common to all three interrupts.



5.4.6 Interrupt Timing

I/O clock (IOCL-) is a free-running 833 KHz clock, with the frequency altered under two conditions:

1. During interrupt recognition and interrupt instruction execution, IOCL- is frozen in the high state;
2. During memory and I/O cycles (input, output, sense or control), the "on" state of the clock (i.e., the low level) is stretched, effectively increasing the clock period for that one period. For I/O cycles, IOCL- period is stretched by 600 nanoseconds. For memory cycles, IOCL- period is stretched an amount which is a function of memory access time, typically 600 nanoseconds.

Timing for the interrupt address request cycle is analogous to input timing (paragraph 5.3.1); signal IAR- replaces IN- in this case.

Signal IOCL- is used in peripheral controllers to synchronize IUR- interrupts to the machine. IOCL- is turned off during the interrupt processing to prevent priority resolution ambiguities.

Interrupt acknowledge (IUA-) is true (low) during the interrupt address request cycle and subsequent interrupt instruction execution. IOCL- is held in the "off" state for that period.

5.4.6.1 IL1 and IL2 Timing

Figure 5-3 provides the timing of all signals for IL1 and IL2 interrupts. During IAR time (interrupt address request), no interrupt address need be provided to the Processor. If an address is provided, it will automatically be logically "OR'ed" with the dedicated address (: 0002 and : 0006 for IL1 and IL2, respectively).

5.4.6.2 IUR Timing

Figure 5-3 also provides all timing requirements for IUR interrupts. During IAR time, the interrupting controller with priority must provide an interrupt address (otherwise, an interrupt address of : 0000 will be assumed) within the timing constraints shown.

No peripheral controller may change the state of its PROT- line during the period that interrupt acknowledge (IUA) is true; otherwise a priority resolution ambiguity will occur.

Priority ripple time down the priority string is a function of the complement of peripheral controllers installed in the system. Refer to the ALPHA 16 LSI and NAKED MINI 16 LSI I/O Interface Design Guide for a complete discussion of priority ripple time and priority logic implementation constraints.

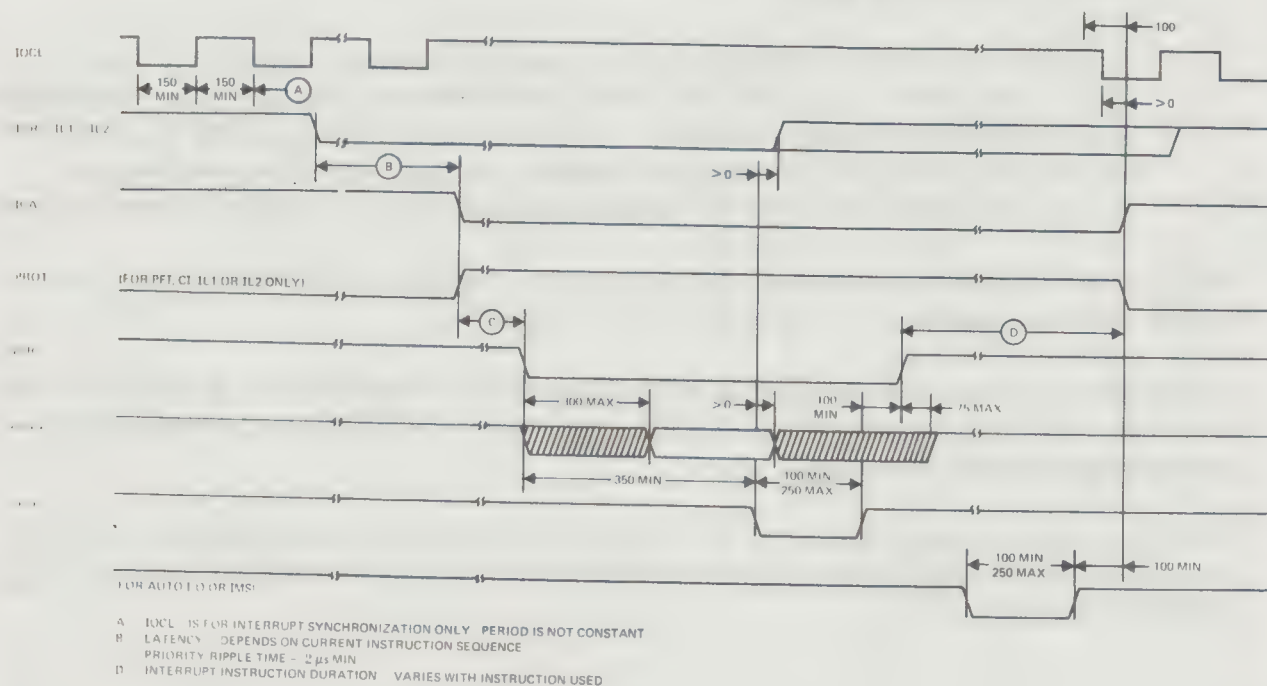


Figure 5-3. Interrupt Transfer Timing

5.5 DIRECT MEMORY ACCESS (DMA)

The ALPHA LSI system allows for specially designed I/O controllers (referred to as DMA controllers) to communicate directly with main memory independent of the operating program. DMA operation alters none of the Processor registers or status flip-flops.

The DMA controller is responsible for controlling the data transfer between memory and the controller. For DMA operation, the DMA controller stops the Processor to acquire the external bus, performs the data transfer, then releases the Processor so the operating program may continue. Processing throughput is inversely related to DMA execution time.

DMA operation takes precedence over any Processor operation (except Reset) including all interrupt and console operations. For power fail conditions (as indicated by signal PFD-), it is the responsibility of the DMA controller to release the Processor so that the Processor may execute the power-down routine.



5.5.1 Multiple DMA Controllers

The ALPHA 16 LSI system can accommodate a number of DMA controllers actively competing for service simultaneously. The DMA priority string, as discussed in paragraph 5.2.3.4.1, is used to resolve priority among several controllers. The number of DMA controllers which may be used within the system is limited only by priority-ripple time. Where more than 200 nanoseconds is required for priority-ripple time, each DMA controller must be designed to abstain from beginning the memory cycle until sufficient time has elapsed for priority-ripple.

Priority resolution occurs at two times: after the leading edge of SACK- and after the trailing edge of MACK-. If only one DMA controller is installed in the system, or if only one DMA controller is allowed to be active at a time in multiple DMA configurations, then no priority-ripple time need be allocated.

5.5.2 DMA Controller Slots

DMA controllers must be installed in the 200-series slots if priority resolution among several DMA controllers is required. The DMA priority string is not available at connector pins of the 100-series slots.

5.5.3 Latency, Data Transfer Rate

Latency is defined with respect to DMA operation as the elapsed time between the leading edge of STOP- until the leading edge of SACK-. For the ALPHA LSI system, maximum latency is 3.6 microseconds. Data transfer rate is strictly a function of memory cycle time. In the standard configuration, a data transfer rate of 625,000 words/second may be achieved with the 1.6 microsecond cycle-time memory.

5.5.4 Memory Overlap

Data transfer rates of up to 1,250,000 words/bytes per second can be achieved by the DMA Controller performing successive memory cycles from different memory banks and exploiting the memory overlap capability of the computer.

5.6 ELECTRICAL CHARACTERISTICS

The external bus is best classified as a hybrid tri-state open-collector (wire-OR) bus, unterminated. Most Processor drivers are tri-state power elements, capable of sinking 32 mA at 0.4 Vdc maximum and sourcing 2.0 mA at 2.4 Vdc minimum. In a few isolated cases, open-collector TTL drivers (32 mA sink at 0.4 Vdc) are used. Processor receivers present one standard TTL load to the line (-16 mA at 0.8 Vdc, 40 A at 2.4 Vdc). Depending on the nature of the particular signal, pullup resistors to +5 are used.



Open-collector drivers in I/O and memory modules are permitted on those bus lines for which pullup resistors are provided. Minimum required drive capability is -32 mA at 0.4 Vdc max. Tri-state drivers electrically equivalent to the Processor bus drivers are also allowed, as long as the logic design of the system guarantees that no two tri-state drivers connected to the same bus line are simultaneously enabled. Receivers on I/O and memory modules may be any standard 74 series TTL device. Only one such receiver per module is permitted. Maximum loading shall not exceed 1.8 mA per module.

Logic Levels

(Negative-true)

logic "1": +0.4 Vdc max.
logic "0": +2.4 Vdc min

Table 5-1 summarizes the driver, receiver and pullup circuits. Table 5-2 summarizes the slot A100 special signals and their termination requirements.

5.7 MOTHERBOARD ORGANIZATION

Any slot (other than the slot dedicated to the NAKED MINI 16 LSI card) can accept either an I/O or memory module.

Figure 5-2 provides a diagram of the system motherboard. The motherboard provides for six slots used as follows:

Slot	Purpose
A	NAKED MINI 16 LSI Computer
B	Universal (Memory or I/O)
C	Universal (Memory or I/O)
D	Universal (Memory or I/O)
E	Universal (Memory or I/O)
F	Power Supply

In any given slot, either a full-width card (15" x 16.5") or two half-width cards (each 7.5" x 16.5") may be installed. One slot contains two connectors. The pins on one connector of a slot are numbered 100 through 186, and is referred to as the 100-series connector; similarly, the other connector (pins 200 through 286) is referred to as the 200-series connector.

With the exception of the priority chains, memory bank control, and two special CPU power supply signals, all signals are wired in a U fashion through all half-card connectors. All exceptions are described below.



Table 5-1. External I/O Bus Load, Drive and Termination Summary

		DEVICE TYPE(S) (REFER TO NOTE 1)					
SIGNAL	PIN	CPU	MEMORY	I/O CONT	DMA CONT	CONSOLE	OPT. BD
GND	1						
+12V	2						
+12V	3						
+12V	4						
+12V	5						
+12V	6						
+12V	7						
-12V	8						
DPIN-	9						
DPOT-	10						
NOTE 2	11						
NOTE 2	12						
NOTE 3	13						
+5V	14						
+5V	15						
MST-	16						
MACK-	17						
RD-	18						
PCLK-	19						
SLB-	20						
PFD-	21						
MDIS-	22						
AB06-	23						
AB09-	24						
AB10-	25						
AB11-	26						
GND	27						
GND	28						
AB12-	29						
AB13-	30						
AB14-	31						
AB15-	32						
NOTE 3	33						
NOTE 3	34						
STOP-	35						
SACK-	36						
NOTE 2	37						
NOTE 2	38						
MBOT-	39						
DB00-	40						
DB01-	41						
DB02-	42						
DB03-	43						
+5V							

		DEVICE TYPE(S) (REFER TO NOTE 1)					
SIGNAL	PIN	CPU	MEMORY	I/O CONT	DMA CONT	CONSOLE	OPT. BD
+5V	44	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB04-	45	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB05-	46	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB06-	47	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB07-	48	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB08-	49	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB09	50	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB10-	51	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB11-	52	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB12-	53	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB13-	54	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB14-	55	1.5, 6	1.5	2.5	1.5	1.5	1.5
DB15-	56	1.5, 6	1.5	2.5	1.5	1.5	1.5
EXEC-	57	1.6	1.5	2.5	1.5	1.5	1.5
IN-	58	1.6	1.5	2.5	1.5	1.5	1.5
GND	59	1.6	1.5	2.5	1.5	1.5	1.5
GND	60	1.6	1.5	2.5	1.5	1.5	1.5
IOCL-	61	1.6	1.5	2.5	1.5	1.5	1.5
OUT-	62	1.6	1.5	2.5	1.5	1.5	1.5
CLK-	63	3	1.5	2.5	1.5	1.5	1.5
SER-	64	5.6	1.5	2.5	1.5	1.5	1.5
IUR-	65	5.6	1.5	2.5	1.5	1.5	1.5
IL1-	66	5.6	1.5	2.5	1.5	1.5	1.5
IAR-	67	1.6	1.5	2.5	1.5	1.5	1.5
IL2-	68	5.6	1.5	2.5	1.5	1.5	1.5
EST-	69	2.5, 6	1.5	2.5	1.5	1.5	1.5
IUA-	70	1.6	1.5	2.5	1.5	1.5	1.5
PLSE-	71	1.6	1.5	2.5	1.5	1.5	1.5
ECHO-	72	1.6	1.5	2.5	1.5	1.5	1.5
+5V	73	1.6	1.5	2.5	1.5	1.5	1.5
+5V	74	1.6	1.5	2.5	1.5	1.5	1.5
AB00-	75	1	5	5	1.5	5	5
AB01-	76	1	5	5	1.5	5	5
AB02-	77	1	5	5	1.5	5	5
AB03-	78	1	5	5	1.5	5	5
AB04-	79	1	5	5	1.5	5	5
AB05-	80	1	5	5	1.5	5	5
AB06-	81	1	5	5	1.5	5	5
AB07-	82	1	5	5	1.5	5	5
PRIN-	83	1	5	5	1.5	5	5
PROT-	84	4	5	5	1.5	5	5
GND	85	4	5	5	1.5	5	5
GND	86	4	5	5	1.5	5	5

NOTES: 1. DEVICE TYPES ARE AS FOLLOWS:-

1) TRI-STATE DRIVER

2) 32 MA OPEN-COLLECTOR DRIVER (7439 OR EQUIV.)

3) 32 MA TTL DRIVER (7437 OR EQUIV.)

4) 16 MA TTL DRIVER (7400 OR EQUIV.)

5) TTL RECEIVER (7404 OR EQUIV.)

6) PULL-UP RESISTOR (1 KOHM)

J) JUMPER

9) STRAIGHT THRU SIGNAL (NO DEVICES IN SIGNAL PATH)

2. DPIN-, DPOT-, MBIN- AND MBOT- ARE STRUNG THROUGH THE 200 SERIES CONNECTORS ONLY. THESE PIN POSITIONS ARE UNASSIGNED ON THE 100 SERIES CONNECTORS AND ARE RESERVED FOR FUTURE EXPANSION.

3. THESE PINS ARE UNASSIGNED ON BOTH THE 100 AND 200 SERIES CONNECTORS AND ARE RESERVED FOR FUTURE EXPANSION.

Table 5-2. Slot A100 Special Signals

SIGNAL	PIN	CPU	SOURCE/DESTINATION
SSW-	9	5,6	CONS 2
IF-	10	2,6	CONS 5
120Hz	11	5	P.S.
+5H	12		P.S.
AL-	33	5,6	CONS 2
BM-	34	2,6	CONS 5
OV-	37	2,6	CONS 5
START-	38	2,5,6	CONS 2
SERV-	83	2,5,6	CONS 2
CINT-	84	5,6	CONS 2

NOTE 1. All other pin assignments are the same as Universal slot.

2. Device types are as follows:

2. = 32ma open-collector drives (7438 or equivalent)

5. = TTL receiver (7400 or equivalent)

6. = Pullup resistor (1 Kohm)

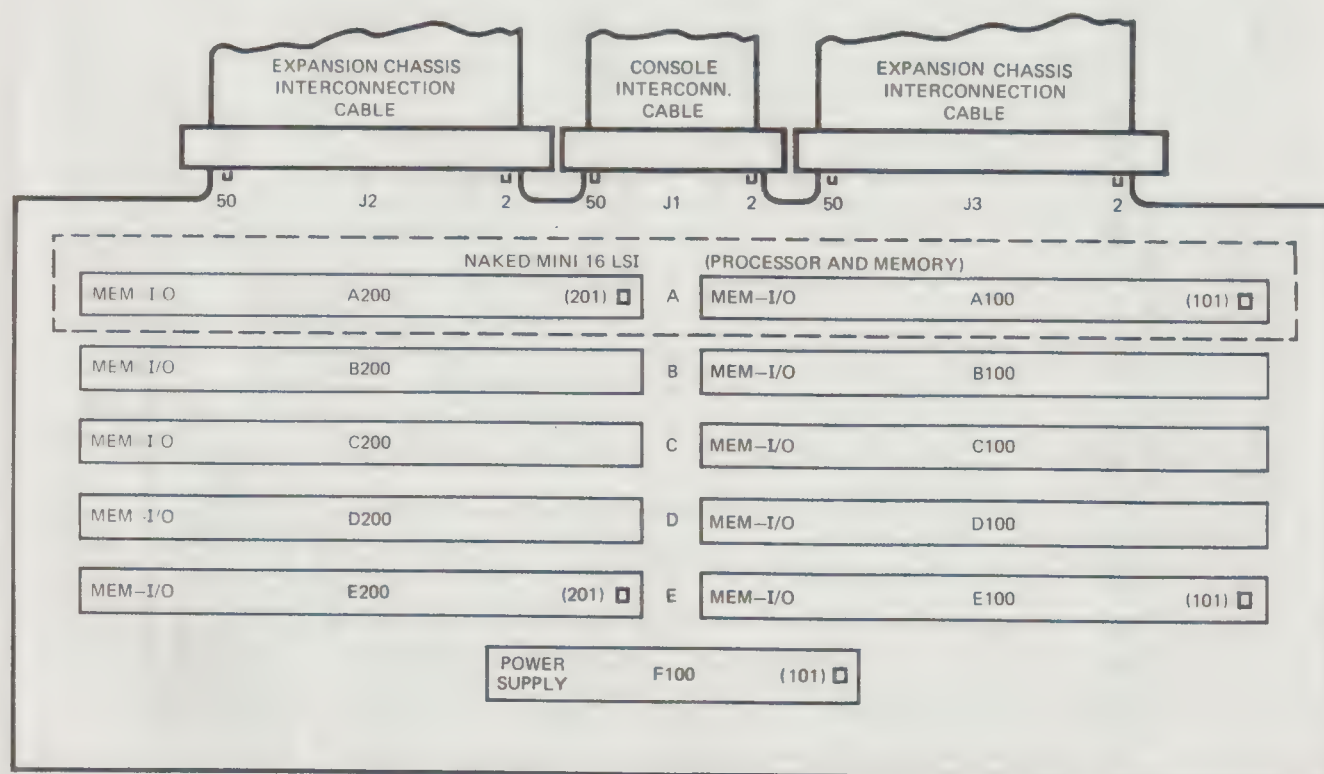


Figure 5-4. ALPHA 16 LSI Motherboard Slot Organization (Rear View)



5.7.1 Interrupt Priority

The daisy-chained interrupt priority string (PRIN-, PROT-) is wired in S fashion beginning at the 100-series connector of slot A, across to the 200-series connector, then in reverse direction across the two B slot connectors, etc., until all slots are connected. Both ends of the chain are connected to the expansion connectors. Both PRIN- and PROT- on the CPU connector (100-series connector, slot A) are used to carry special signals to the console; the actual origin of the priority chain is therefore the memory side of the top slot (slot A100).

5.7.2 Memory Bank Control, DMA Priority

The memory bank control (MBIN, MBOT) and DMA priority (DPIN-, DPOT-) daisy-chains run down the 200-series connectors only. Therefore, memories and DMA controllers must be either full cards or half cards installed on the 200 series side only.

5.7.3 CPU Power Supply Signals

Two lines from the power supply, TTLF (twice the line frequency) and +5 Hz are wired directly between the power supply slot and CPU slot (slot A100).

5.8 EXPANSION

To facilitate expansion of the computer system beyond the first chassis and to provide for interconnect to the ALPHA 16 LSI console, expansion connectors are supplied on the motherboard immediately above slot A. Two connectors are provided for I/O bus expansion and one connector is provided to interconnect the console.

In the event that insufficient slots are provided in the basic ALPHA 16 LSI chassis for a given application, the external bus may be expanded via a buffer card, cables, and expansion chassis.

The expansion chassis is a second ALPHA 16 LSI chassis (identical motherboard, etc.). A buffer card is required to regenerate the external bus. Cables interconnect the buffer card to the two motherboards via the expansion connectors. The external bus of the computer on the motherboard may not be extended without electrical buffering.

Expansion may extend to two, three or more chassis. As expansion chassis are installed, a speed degradation will occur. Memory modules located in expansion chassis will exhibit an apparent slower system access time. Similarly, I/O modules located in a second expansion chassis or beyond may require that the timing circuit of the computer be altered to provide additional phase stretching during I/O operations. This timing circuit is modified simply by changing an option-jumper connector which configures all jumper-controlled Processor options in the machine. This option-jumper connector mounts to the read-edge of the computer option card.



5.9 NAKED MINI 16 LSI EXTERNAL BUS REQUIREMENTS

In applications where the NAKED MINI 16 LSI computer is used without the system motherboard and is instead connected to I/O and/or memory modules via user-supplied cabling, printed circuit card, etc., the line length of each signal must be limited to 18 inches.

The external bus must be designed to minimize crosstalk, reflections, etc., so as to preserve signal integrity. Recommendations as to line termination are available upon request. In general, consultation with Computer Automation is recommended to ensure system performance.

5.10 TWO-MODULE OPTIONS

Any option requiring more than one printed circuit card may not use the motherboard for interconnection. Unique interconnections may be made via a jumper cable installed on the rear edge of the two cards.



Section 6

CONSOLE

6.1 GENERAL

The system console provides the user with a direct control/display interface to the Processor. Except for the inclusion of a few dedicated signals, the console is organized as an I/O device of the computer.

The main components of the console logic are:

1. A 16-bit Console Data register, which interfaces to the Console display indicators, Console switches, and computer I/O bus;
2. Control unit which interfaces to the Console display indicators and switches, and provides control signals to the Processor;
3. Four-bit Console Sense register, which shares the Console interface with the Console Data register.

All switches are momentary action "touch" switches with the exception of the 'console disable' which is a slide switch and all indicators are light emitting diodes.

6.2 CONSOLE ORGANIZATION (Figure 6-1)

The displays and switches of the console are organized into four primary groups:

1. Data Display and Entry. This group provides the switches and displays which interface to the Console Data register and the Console Sense register, and determines whether or not the Console Data register or Console Sense register is connected to the console displays and switches.
2. Register Select. This group defines the source or destination of data displayed or entered, respectively, in the Data Display and Entry group, and controls the direction of data transfer (read vs write).
3. Control. This group includes the various control indicators and switches for operating the computer.
4. Status. This group provides displays for various status conditions of the system.



6.3 FUNCTIONAL DESCRIPTION

A detailed description of the functions of the various displays and switches, by group, is provided in table 6-1.

Table 6-1. Console Switches and Indicators

SWITCH OR INDICATOR	PURPOSE
<u>System Status Section</u>	
ON Indicator	On when power is applied, off when power is removed. The main power switch is located on the rear of the computer.
ENABLE Slide Switch and Indicator	The console enable/disable slide switch is located in a recess on the edge of the console. When the switch is on, the ENABLE indicator is on. Likewise, when the switch is off the indicator is off. When in the ENABLE state all switches and indicators are enabled. When in the disabled state the only functions that are effective are: 1. The SENSE switch and indicator 2. The Console Sense register, Register Display indicator and hex entry keyboard when the SENSE/DATA indicator is on.
BYTE Indicator	On when the Processor is in Byte Mode. Off when the Processor is in Word Mode.
OV Indicator	On when the Processor overflow flip-flop is on. Off when the overflow flip-flop is off.
SENSE Switch and Indicator	The SENSE Switch toggles the SENSE Indicator. The SENSE Indicator may be tested by program instructions. The Sense test will be true if the SENSE Indicator is on.
<u>System Control Section</u>	
STOP Switch and Indicator	The STOP Switch toggles the STOP Indicator. The Indicator is on when the Stop Mode is established. When the indicator is off the Run Enable Mode is established. When the Stop Mode is established and the console is enabled (ENABLE indicator on), data entry and display operations may be performed. In addition, the Processor will fetch and execute one program instruction each time the RUN switch is pressed.

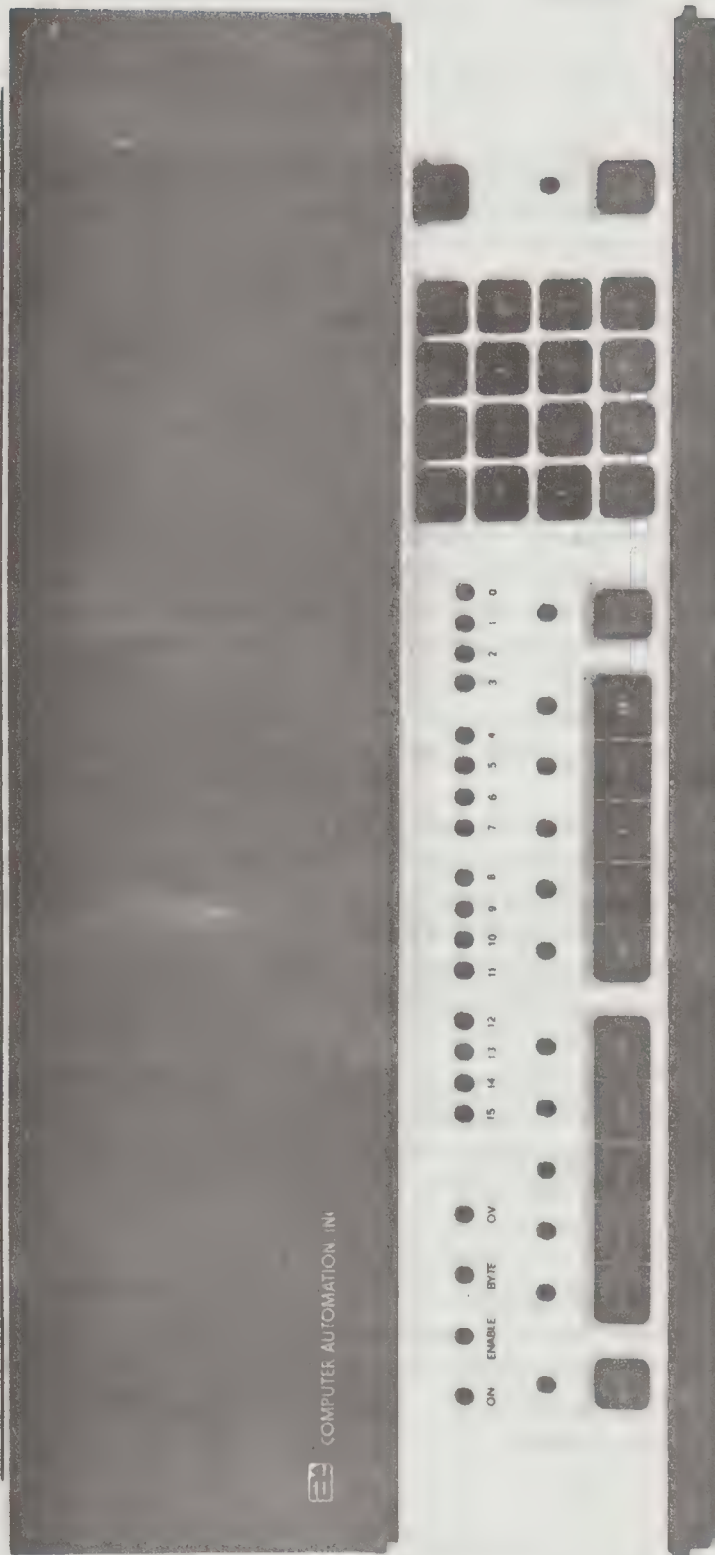


Figure 6-1. ALPHA 16 LSI Console



Table 6-1. Console Switches and Indicators (Cont'd)

SWITCH OR INDICATOR	PURPOSE
RESET Switch and Indicator	When in the Run Enable Mode, data entry and display operations may not be performed and the Run Mode is enabled but not entered until the RUN switch is pressed. The indicator is on when the RESET switch is on and remains on only as long as the switch is pressed. The RESET switch generates a system reset signal which causes the Processor and all interfaces to be initialized. The RESET switch should not normally be used to stop the computer. If RESET is pressed while the computer is running, the instruction currently being executed may not complete. The STOP switch should normally be used to halt the computer. The only time that RESET should be used to halt the computer is in the case where the Processor is hung up in a non-escapable one instruction loop (e.g., multi-level indirect address instruction with closed address chain).
AUTO Switch and Indicator	The AUTO switch is used to initiate an Autoload sequence if the Autoload option is installed. The AUTO switch is enabled only during the Run Enable Mode. Depressing the switch establishes the Run Mode and initiates the Autoload sequence. The indicator turns on when the switch is pressed and remains on until the Autoload sequence is completed.
INT Switch and Indicator	The INT switch is used to initiate a Console Interrupt. The switch is enabled only during the Run Mode. The indicator turns on when the switch is pressed and remains on until the Processor honors the Console Interrupt Request.
RUN Switch and Indicator	The RUN switch is used to establish the Run Mode when the STOP indicator is off. When the STOP indicator is on, the RUN switch causes one instruction to be fetched and executed when pressed. The WRITE indicator is turned off whenever RUN is pressed. The RUN indicator is turned on when in the Run Mode.
<u>Entry/Display Section</u>	
Register Display Indicators (0 thru 15)	The 16 Register Display Indicators display the contents of either the Console Data register or the Console Sense register depending on the state of the SENSE/DATA indicator. When the SENSE/DATA indicator is off, the contents of the Console Data



Table 6-1. Console Switches and Indicators (Cont'd)

SWITCH OR INDICATOR	PURPOSE
<u>Entry/Display Section</u>	
Register Display Indicators (0 thru 15)	The 16 Register Display Indicators display the contents of either the Console Data register or the Console Sense register depending on the state of the SENSE/DATA indicator. When the SENSE/DATA indicator is off, the contents of the Console Data register are displayed. The Console Data register contains either: 1) the most recent contents of the A, X, I or P register or Memory as requested by the Register Select switches; 2) the last Processor output to the Console Data register; or 3) the last keyboard entry to the Console Data register. When the SENSE/DATA indicator is on, the contents of the 4-bit Console Sense register are displayed on the Register Display Indicators. The Console Sense register contains either the last keyboard entry to the Sense register or the last Processor output via the Status Output Command. The upper 12 Register Display indicators are turned off when displaying the Console Sense Register.
Register Select Switches and Indicators (A, X, I, P and M)	The five Register Select switches determine which one of four Processor registers or memory is to be involved in a read/write operation. Each switch has a corresponding indicator which turns on when a given switch is pressed. The indicators are interlocked such that only one indicator is on at a time. The A, X, I and P switches cause a transfer to occur between the target register and the Console Data register. The M switch causes a transfer between the memory and Console Data register to occur and also causes the P counter to increment after the transfer. This feature permits manual scanning or loading of sequential memory locations by repeated pressing of the M switch.
READ/WRITE Switch and Indicator	The READ/WRITE switch is used in conjunction with the Register Select switches. When the READ/WRITE indicator is on, the contents of the Console Data register will be written into the target register or memory when the appropriate Register Select switch is pressed. When the READ/WRITE indicator is off, the contents of the selected register or memory are copied into the Console Data register and displayed.



Table 6-1. Console Switches and Indicators (Cont'd)

SWITCH OR INDICATOR	PURPOSE
Hexadecimal Entry Keyboard (0 thru F)	The Hexadecimal Entry Keyboard consists of 16 switches which are used to enter data into either the 16-bit Console Data register or the 4-bit Console Sense register as determined by the SENSE/DATA switch and indicator. When the SENSE/DATA indicator is off, each depression of a key causes a corresponding 4-bit binary hex code to be entered into the four least-significant bits (LSB's) of the Console Data register with the previously entered data shifted four places to the left. The Console Data register will be statically displayed as long as the SENSE/DATA indicator is off and the computer program does not alter the contents of the Console Data register. When the SENSE/DATA indicator is turned on, each depression of a hex entry key causes the corresponding binary hex code to be entered into the four-bit Console Sense register. The Console Sense register is statically displayed in the four least significant Register Display indicators so long as SENSE/DATA is in the on state and the computer program does not modify the contents of the Console Sense register. The upper 12 Register Display indicators are extinguished.
SENSE/DATA Switch and Indicator	The SENSE/DATA switch toggles the SENSE/DATA indicator which determines whether the Console Data register or the Console Sense register is to be connected to the hex entry keyboard and the Register Display indicators. If the SENSE/DATA indicator is off, the hex entry keyboard is used to enter data into the Console Data register and the Register Display indicators are connected to the Console Data register. If the SENSE/DATA indicator is on, the keyboard and display are connected to the Console Sense register.
CLEAR Switch	The CLEAR switch, when pressed, clears data from the Console Data register. The switch does not affect the Console Sense register.



6.4 PROGRAMMED OPERATIONS

The console, being an I/O device of the Processor, has associated with it a set of I/O instructions. These are standard I/O instructions of the Processor, with device address zero and special function codes used to control the particular function.

6.4.1 Data Register Instructions

Any select and present instruction with device address zero and function code four will load the Console Data register with the output data. Any unconditional input instruction (e.g., INA) with device address zero and function code four will input data from the Console Data register to a Processor register.

6.4.2 Program Status Word

The Program Status Word used internal to the Processor for byte mode, interrupt enable and overflow status also has five bits within it associated with console status. The Program Status Word is either input or output to/from a Processor register via an unconditional input or output instruction using device address zero and function code zero. The format for the Program Status Word is:

Bit	Function*
0	Overflow (OV)
1	Interrupt Enable (EIN)
2	Byte Mode (BYTE)
3	SENSE Indicator
4	Sense Register, bit 0
5	Sense Register, bit 1
6	Sense Register, bit 2
7	Sense Register, bit 3

* Functions for bits 0, 1 and 2 are independent of the console, dependent only on internal Processor functions.

When a status word input instruction is executed, the bits of a Processor register are set in accordance with its associated function. When a status word output instruction is executed, the SENSE Indicator and Sense register are updated in accordance with the Processor register being output.

6.4.3 Console Control Word

The Console Control Word (CCW) is input to the Processor when any unconditional input instruction with device address and function code zero is executed simultaneous with signal IF- being true, as such, the CCW is not accessible to the programmer.



A CCW may be executed as an instruction. If done, a data transfer between the console and the Processor or memory will take place and the Processor will then halt.

The format for the CCW is as follows:

<u>Bit</u>	<u>Function</u>
0	Input/Output control. If set, an output operation (data transfer from Processor to console) will take place.
1	A Register Select. If set, the Processor A register will be a source or destination register;
2	X Register Select. If set, the Processor X register will be a source or destination register;
3	I Register Select. If set, the Processor I register will be a source or destination register.
4	P Register Select. If set, the Processor P register will be a source or destination register.
5	Memory Location Select. If set, a location in memory will be a source or destination register. The memory location is defined by the Processor P register. Furthermore, if a transfer between memory and the console is executed, the P register will be incremented.

Bits 1 through 5 in the CCW are mutually exclusive: only one bit may be set for proper operation.

6.5 CONSOLE - PROCESSOR INTERFACE

The Console interfaces to the Processor via the external bus, plus special control lines not generally considered to be part of the external bus. The special lines and the associated functions are described below:

SERV- Console Service. The SERV- signal (negative true) is issued by the console to command the Processor to service the console. The SERV- line may be considered to be an interrupt line with priority over all interrupts, but superseded by DMA operation. The Processor responds to SERV- by performing a CCW input (actually, an instruction fetch from the console instead of memory). The CCW determines the required servicing.



- IF-** Instruction Fetch. The IF- signal (negative true), issued by the Processor, envelops the instruction fetch cycle. In response to SERV-, the Processor performs an instruction fetch cycle, which in this case is a CCW fetch instead of the usual memory read cycle. The console uses IF- to differentiate the CCW input cycle from a status word input cycle; both use device address and function code zero.
- If SERV- is issued coincident with the leading edge of IF- or later, the instruction fetch cycle will cause an instruction to be accessed from memory and subsequently executed before SERV- will be honored. SERV- must lead IF- by at least 1.5 microseconds to guarantee that the next IF- cycle will be a CCW input cycle.
- START-** Start Processor. Signal START- (negative true) is issued by the console to command the Processor to resume processing. START- must be a minimum of 1.5 microseconds wide. The Processor resumes processing on the trailing edge of START-. Signal SERV- must precede the trailing edge of START- by at least 1.5 microseconds to guarantee that the Processor will immediately perform a CCW input instead of a memory read cycle when Processing is resumed.
- CINT-** Console Interrupt. CINT- (negative true) is issued by the console to interrupt normal processing. Signal CINT-, once issued, must be held true until signal IUA- (Interrupt Acknowledge) is true.
- SSW-** Sense Switch. Signal SSW- (negative true), issued by the console, tracks the console Sense switch. No synchronization is required. If the Sense switch is set, signal SSW- is true.
- AL-** Autoload. Signal AL- (negative true) is issued by the console to command the optional autoload logic to perform an autoload sequence. The autoload sequence is initialized on the leading edge of AL- and commences on the trailing edge of AL-. The AL- pulse width must be 100 nanoseconds minimum.
- OV-** Overflow. The OV- signal (negative true) is issued by the Processor. OV- tracks the Overflow flip-flop internal to the Processor.
- BM-** Byte Mode. The BM- signal (negative true) is issued by the Processor. BM- tracks the Byte Mode flip-flop internal to the Processor.



6.6 PHYSICAL INTERCONNECT

The console interconnect to the Processor is accomplished via a cable and the motherboard. A cable connects the console PC board with the system motherboard. The Processor receives these signals via the motherboard - CPU connector.



Section 7

OPTION CARD

7.1 GENERAL

The option card provides the Teletype/Modem/CRT controller interface (TTY/CRT), Real Time Clock (RTC), Autoload (AL), and an option select connector for the selection of various options within the computer system.

This section provides minimum hardware specifications of the above. For detailed operating procedures and programming considerations, refer to the appropriate interface description.

7.2 TELETYPE/CRT CONTROLLER

7.2.1 Functional Description

The TTY/CRT Controller option interfaces the computer in either a half duplex or a full duplex mode to one of the following devices:

ASR-33 or -35 teletypewriter utilizing 20 mA current loops,

Alphanumeric CRT or other devices utilizing EIA RS232 logic levels.

7.2.2 Transfer Rates

Available data transfer rates are as follows:

<u>Bit Transfer Rate</u> (Baud)	<u>Byte Transfer Rate</u> (Characters per Record)
75	6.82
110	10
134.5	12.12
150	13.64
300	27.27
600	54.54
1200	109.09
2400	218.18
4800	436.36
9600	872.72



The specific rate is determined by a jumper wire on the option jumper connector. The absence of jumper wires results in selection of the standard TTY transfer rate of 110 baud.

7.2.3 External Logic Levels

Interface logic levels to/from the external devices are as follows:

To/from TTY: True = current flow (20 mA current loop)
 False = absence of current

To/from CRT: True = -9 Vdc
 False = +9 Vdc

7.2.4 Interrupt Locations

Standard interrupt locations are as follows:

Half-duplex

Address : 0002 Word Interrupt. The controller interrupts to the location when the controller Word Transfer Mask is set, interrupts are enabled, and the Buffer Ready flip-flop is set.

Address : 0006 End-of-Block Interrupt. The controller interrupts to this location when the Block Transfer Mask is set, interrupts are enabled, and an ECHO signal is received from the computer indicating that the last word of a block has been transferred.

Full Duplex

Address : 0022 Word Interrupt, input to computer.

Address : 0026 End-of-Block Interrupt, input to computer.

Address : 0002 Word Interrupt, output from computer.

Address : 0006 End-of-Block Interrupt, output from computer.



7.2.5 Alternate Interrupt Locations

In the case of half-duplex communication only, the interrupt addresses of : 0022 and : 0026 may be substituted for interrupt address : 0002 and : 0006, respectively. This option is jumper-selected at the option jumper connector.

7.2.6 Interrupt Location Offset

By selecting the appropriate jumper on the option jumper connector, all interrupt locations may be offset by : 0100 locations. This applies as well to the half-duplex alternate interrupt locations.

7.2.7 I/O Instructions

The TTY/CRT controller is assigned a device address of seven. The following paragraphs summarize the I/O instructions which apply to the controller by referencing the function code of the I/O instruction.

7.2.7.1 Select Commands

<u>Function Codes</u>	<u>Description</u>
0	Enable Auto Echo
1	Keyboard input mode (CRT)
2	Step Read (TTY)
3	Continuous Read (TTY)
4	Initialize Controller (half-duplex mode)
4	Initialize Controller for full duplex mode (SEA,SEX) A/X = : 0001.
5	Set Word Transfer Mask (half-duplex mode)
5	Set Word Transfer Mask (full duplex mode; SEA,SEX,A/X = : 0001) (must be in full duplex mode).
6	Set Block Transfer Mask (half-duplex mode)
6	Set Block Transfer Mask (full duplex mode; SEA,SEX,A/X = : 0001) (must be in full duplex mode).
7	Reset Word and Block Transfer Masks.

7.2.7.2 Sense Commands

<u>Function Code</u>	<u>Description</u>
0	Input data request (full duplex only)
1	Buffer Ready (applies only to output data request during full duplex operation).



Sense Commands (Cont'd)

<u>Function Code</u>	<u>Description</u>
2	Word Transfer Mask on (applies to output mask only in full duplex mode).
3	Controller not busy
4	CTS signal on (CRT)
5	Motor on (TTY)
6	Parity or framing error did not occur on last input.
7	Input Word Transfer Mask set (full duplex only).

7.2.7.3 Input Commands

<u>Function Code</u>	<u>Description</u>
0	Input Byte of data from controller
1	Sense buffer ready flip-flop and input byte of data on true response.

7.2.7.4 Output Commands

<u>Function Code</u>	<u>Description</u>
0	Transmit data
6	Turn motor on (TTY). Turn on RTS signal (CRT).
7	Turn motor off (TTY). Turn off RTS signal (CRT).

7.3 REAL TIME CLOCK

7.3.1 Functional Description

The Real Time Clock (RTC) generates two interrupts based on a clock source, and is used to measure elapsed time or create a time-of-day clock for the computer. The RTC option keeps time by counting pulses of a known frequency.



7.3.2 Clock Sources

The user may select the clock source via the option connector at the rear edge of the option card:

1. 10 kHz (100 microsecond period)
2. 1.0 kHz (1.0 millisecond period)
3. 100 Hz (10 millisecond period)
4. Double the AC line frequency (typically 120 Hz, or 8.33 millisecond period)
5. External frequency standard, maximum frequency of 100 kHz (connected to rear edge connector of option card)

The first three sources are generated from the computer 20 MHz crystal oscillator. With no option jumper installed, the 100 Hz source is automatically selected.

7.3.3 Interrupts

The IUR interrupt line is used for two interrupts:

1. Clock interrupt (address vector : 0018):
Clock interrupts are generated once per cycle of the clock source.
2. Sync interrupt (address vector : 001A):
Sync interrupts are generated as a result of an ECHO condition occurring under a clock interrupt, i.e., when a count value is incremented to zero in the case where an IMS instruction is used as an interrupt instruction for the Clock interrupt.

The interrupt address vectors may be offset by : 0100 locations via installation of the appropriate jumper on the rear edge connector of the option card.

7.3.4 I/O Instructions

The RTC option is controlled by a number of I/O Select instructions, all using device address eight. These instructions are summarized below:

<u>Instruction</u>	<u>Description</u>
SEL 8,0	Enable RTC clock and sync interrupts
SEL 8,2	Arm sync interrupt
SEL 8,3	Clear RTC clock and sync interrupts
SEL 8,4	Initialize and reset RTC
SEL 8,7	Disarm sync interrupt



7.4 AUTOLOAD

7.4.1 Functional Description

The Autoload option provides for the automatic loading of a binary program into computer main memory. The autoload sequence consists of Processor execution of a binary loader program stored in 256 words of read-only-memory (ROM) provided with the Autoload option.

The autoload sequence is initiated by any of the following:

1. by depressing the console AUTO switch in a system configuration which includes a console;
2. by momentarily grounding a pin on the motherboard connector provided for the console, in configurations where the standard console is not used (the same signal is available at the NAKED MINI 16 LSI connector in NAKED MINI applications);
3. by momentarily grounding a pin on the rear edge connector of the option card, for remote autoload applications.

7.4.2 The Autoload Sequence

When the autoload sequence is initiated, a system reset (RST-) is generated, thereby placing the Processor in the word mode and disabling interrupts.

ROM address space is : 0000 through : 00FF. The first instruction is accessed from ROM location : 0000. The autoload sequence is terminated with the execution of SEL : 00 which disables the autoload ROM after execution of one more instruction. In this case the instruction must be accessed from the ROM.

ROM is accessed during the autoload sequence for instructions only. During instruction fetch, AL logic disables main memory. ROM access time is 1.5 microseconds, maximum. As a result, instruction fetch phases of the Processor are stretched by 1.8 microseconds maximum.

7.4.3 Programmed ROM Entry

AL logic provides for programmed enabling of the ROM. When enabled under program control the ROM appears in memory at address locations : 7800 through : 78FF. All memory access to these addresses will result in a ROM access until the ROM is disabled.

Execution of SEL : 01 will enable the ROM, while, as in the autoload mode, execution of SEL : 00 disables the ROM.



7.4.4 Standard Autoload Program

The standard autoload program inputs from any of the following:

1. Teletype paper tape reader
2. High-speed paper tape reader
3. Nine-track magnetic tape
4. Cassette tape

The user selects the desired device by entering the appropriate code in the Sense register of the Console.

The Sense register of the Console also controls two other options:

1. load vs verify
2. load absolute vs relocating load

For relocating load, the user enters the starting address in the X register.

The console SENSE Indicator controls whether the Processor will halt after loading is completed as opposed to commencing execution of the newly loaded program.

The standard autoload program provides for checksum and format errors. If error detection occurs with magnetic tape or cassette, rewind and reload commences automatically. For paper tape, the autoload program halts should an error be detected; the autoload sequence must be manually restarted in this case.

The standard autoload program uses locations : 0030 through : 003F of main memory as scratch memory.

7.5 OPTION JUMPER CONNECTOR

A connector on the rear edge of the option card is provided for controlling various control options within the computer system. The jumper options are listed below:

1. Secondary Console Sense register. Grounding these four jumper pins introduces corresponding logic '1' bits in the status word for SIA and SIX instructions and in the Console Sense register word for ISA and ISX instructions. (Used only when no console is installed.)
2. Secondary Console SENSE switch. A ground jumper on this pin simulates the Console SENSE switch in a set state for SIA, SIX and conditional jump instructions. (Used when no console is installed.)
3. Secondary Console interrupt switch. A momentary ground jumper simulates a console interrupt. This jumper option is also available at the TTY interface connector. (OR'd with console interrupt switch.)



4. Interrupt offset, high priority. A ground jumper results in all high priority (power fail, console, trap, IL1 and IL2) interrupt locations being offset by : 0100.
5. Interrupt offset, low priority. A ground jumper results in an interrupt location offset by : 0100 for the IUR interrupts generated by I/O controllers located on the option card (TTY and Real Time Clock).
6. Secondary Autoload switch. A momentary ground jumper simulates the console autoload signal and results in the execution of the autoload sequence. This jumper option is also available on the TTY interface connector. (Jumper is active at all times and will first reset the computer if pressed while the computer is running.)
7. I/O stretch extension. Four pins are provided for programming an extended Processor I/O stretch period in increments of 200 nsec. The maximum I/O stretch period is 3.8 microseconds.
8. Real-time-clock period election. Five jumper pins are provided for selection of 10 kHz, 1 kHz or twice the line frequency as a real-time-clock period in place of the standard 100 Hz frequency.
9. TTY/CRT clock period selection. Eleven jumper pins are provided for selection of the transfer rates defined in paragraph 7.2.2.
10. TTY/CRT control. Six pins provide selection of parity inhibit, parity select, stop bit select and word length select in the TTY/CRT controller.



Section 8

POWER SUPPLY

8.1 GENERAL

The ALPHA 16 LSI power supply provides power for all modules installed in the system chassis, plus either the console or the expansion buffer card.

8.2 INPUT POWER

The supply is designed to operate with an input voltage of from 98 to 127 Vrms or 196 to 254 Vrms, 47 to 63 Hz, single phase. Nominal line voltage is 115 Vrms or 230 Vrms. The transformer has two primaries which may be factory configured for either 115 or 230 volt operation. An output from the transformer primary is supplied to power the computer fans in such a fashion that 115V is supplied independent of the primary input voltage used.

8.3 OUTPUT POWER

The following table lists the power output from the supply.

Voltage	Tolerance*	Current Output		Ripple
		Minimum	Maximum	
+5 Vdc	+2%	3 Amp	20 Amp	50 MV p.p
+5 Vdc**	+5%	0 Amp	0.200 Amp	50 MV p.p
+12 Vdc	+2%	0 Amp	3 Amp	50 MV p.p
-12 Vdc	+2%	0 Amp	9 Amp	50 MV p.p

* For load change of 5% to 100% maximum load and specified input voltage and ambient temperature.

**Used only by power failure detector logic.

8.4 TRANSIENT RESPONSE

The transient response and output filtering is capable of supporting step function load current changes while maintaining output voltage changes of $\leq 1\%$ of the static voltage. The load current changes for each output are described below. The step function may occur at any current level greater than 5% of rated output. Maximum total current including the step function will not exceed the ratings in paragraph 8.3.



<u>Output</u>	<u>Step Function Load Change</u>	<u>Maximum Output Current Response Time</u>
+5V @ 20A	1.0 amp maximum	1.0 μ s
+12V @ 3A	1.0 amp maximum	1.0 μ s
-12V @ 9A	4.0 amp maximum	4.0 μ s

8.5 OVER CURRENT AND THERMAL OVERLOAD PROTECTION

Each of the three output voltages is capable of supporting a continuous short circuit without damaging the power supply in any way. The supply is provided with a thermal cut-out switch to prevent component damage in the event of heat buildup.

8.6 OVER VOLTAGE AND REVERSE VOLTAGE PROTECTION

The +5 Vdc 20A output includes an overvoltage crowbar. The crowbar prevents the +5V output from exceeding +6.5 volts in the event of a power supply failure or accidental application of a higher potential from an external source. Each supply output is provided with circuitry to prevent damage to its load or the supply in the event that one is shorted to another or to ground.

8.7 LOGIC SIGNALS

The power supply furnishes two logic signals to the computer, PFD- and TTLF. The PFD- signal goes to its low state when the AC line voltage drops below a voltage level where the power outputs are in danger of going out of regulation. The TTLF signal is a two times power line rectangular wave.

CHARACTERISTICS OF LOGIC SIGNALS

	<u>High State Output Voltage</u>	<u>Low State Output Voltage</u>	<u>Low State Output I</u>	<u>Pull-up Resistor</u>
PFD-	5.0 + 0.5 Vdc	0 + 0.5 Vdc	20 ma min.	270 ohm
TTLF	5.0 + 0.5 Vdc	0 + 0.5 Vdc	20 ma min.	270 ohm

PFD- is a bilateral line, i.e., it can be driven internal or external to the supply.



8.8 POWERFAIL DETECTION AND SEQUENCING

The power supply generates both power up and power down sequences for the purpose of controlling orderly start up and shut down routines within the computer.

Two outputs are provided:

- | | |
|------------|---------------------------|
| 1. PFD- | Powerfail Detected |
| 2. 5V Hang | Regulated Output (200 ma) |

8.8.1 Up Sequence

A Power Up Sequence is initiated when the last of the three necessary input conditions comes true; i.e., input power is sufficiently high for normal power supply operation, low current 5V supply (5V Hang) is within its tolerance limits and PFD- is in its high state.

Output power application occurs in the following order: The 5V Hang is within 5% of its nominal value before the +5 Vdc 20A voltage is greater than 0.5 Vdc. The +5 Vdc 20A voltage is within 2% of its nominal value before the +12 Vdc and the -12 Vdc reach 10% of their nominal values.

8.8.2 Down Sequence

A Down Sequence is initiated when the input voltage falls below its minimum sustaining level.

The line voltage sensor (PFD- circuitry) is factory adjustable to set the trip point for the PFD- signal to any AC input voltage level in the range of 98 to 108 Vrms. The detection point does not drift more than + 2% from its set value. There is sufficient storage in the regulators to insure regulated output for at least 2 milliseconds after PFD- has gone low (true).

The +5V Hang voltage remains within its specified tolerance until the +12 Vdc and -12 Vdc outputs have dropped to no greater than 10% of their specified values.

8.9 MULTIPLE COMPUTER OPERATION

As many as six computer chassis may be operated in parallel. When the configuration is utilized - the PFD - bus, the 5V Hang bus and the ground bus of each is interconnected. The circuitry associated with these buses are so configured that PFD- will trip for power outage on one or more computer chassis and the 5V Hang remains energized.



8.10 STANDARD POWER REQUIREMENTS

The following table lists the power supply loads presented by the standard modules in the computer system:

	<u>+5 Vdc</u>	<u>+12 Vdc</u>	<u>-12 Vdc</u>
NAKED MINI 16 LSI			
memory active	4.0 A	0.75 A	4.4 A
memory standby	4.0 A	0.40 A	0.8 A
Option Card	1.5 A	0.03 A	0.14 A
Console	2.0 A		



Section 9

PHYSICAL SPECIFICATIONS

9.1 INTRODUCTION

The ALPHA 16 LSI Computer utilizes modular packaging to its fullest extent. All major assemblies are sparable and may be replaced with a minimum of disassembly/assembly effort.

The basic ALPHA 16 LSI computer and the expansion version both use the same chassis in modular form.

9.2 REFERENCE DESIGNATION SYSTEM (Figure 9-1)

The basic reference designators for the modular assemblies use an alpha character from "A" through "J". The connector interface reference designators assume the module designator followed by a "P" designation for plug or a "J" designation for jack or a hard mounted connector such as J - P1; (Console Assembly pigtail plug). The exception to this system is the card module connectors on the motherboard assembly. They assume the prefix of the card module and a basic 100 or 200 for each connector that interfaces with the card.

The following is a tabular list of major subassemblies and their corresponding reference designator:

<u>REF DES</u>	<u>SUBASSEMBLY</u>	<u>LOCATION IN COMPUTER</u>
A	Card Module*	1st Slot from Top
B	" "	2nd
C	" "	3rd
D	" "	4th
E	" "	5th
F	Power Supply Module	6th
G	Motherboard Assembly	Behind Console
H	Fan Housing Assembly	Left Side Chassis
J	Console Assembly	Front of Chassis

*Reference Designator A applies to the NAKED MINI 16 LSI computer module in the ALPHA 16 LSI chassis or the top card slot in an expansion chassis.

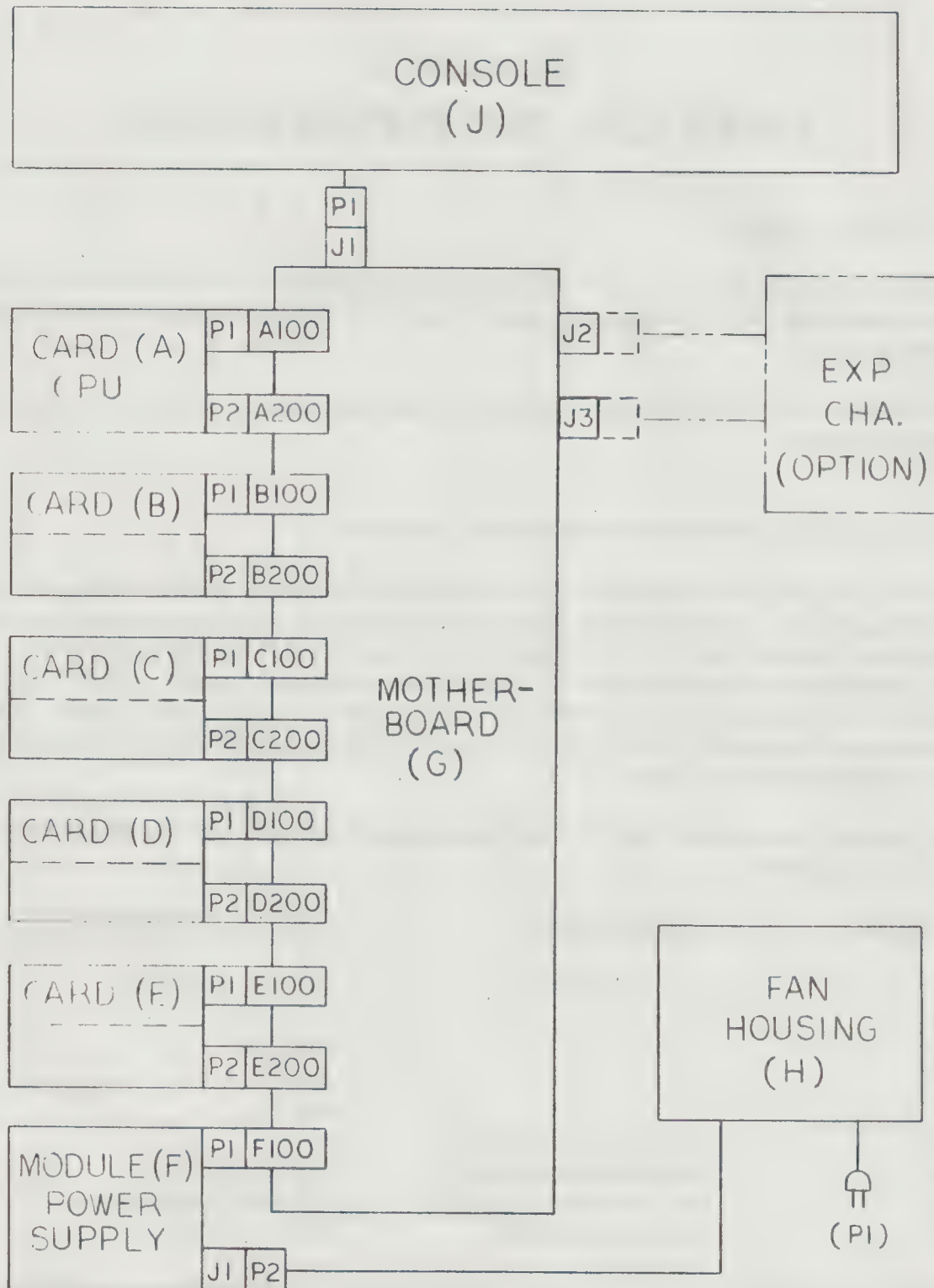


Figure 9-1. ALPHA 16 LSI Reference Designator System



9.3 ALPHA 16 LSI PACKAGING CONCEPT

The ALPHA 16 LSI Computer is made up of a basic chassis to which is assembled a motherboard assembly (M/B), fan housing assembly (F/H), Console assembly and a slide in power supply assembly (P/S). Each chassis is designed to accomodate five slide in printed circuit board card modules which are 15 x 16 inches. All cards are installed in the horizontal plane with the component side up.

The NAKED MINI 16 LSI card module is located on the top slot (A) only. The remaining four slots (B through E) may take any number of option card modules available.

The overall chassis dimensions and mounting parameters are designed to interface with standard cabinets, see figure 9-2.

The motherboard assembly is attached to the chassis from the front with 15 screws.

The fan housing assembly is secured to the side of the chassis with 8 screws.

The power supply assembly slides into the chassis from the rear and is secured via two captive screws from the front (through the motherboard) and three screws at the rear. The fan housing connector (H-P2) is plugged into the power supply through the motherboard to connector F-J1.

The Console assembly snaps onto the front of the chassis. A pigtail cable (J-P1) connector from the console mates with the motherboard at G-J1. The cable is 12" long which provides a service loop for access to the motherboard, fan housing connector and/or power supply screws without disconnecting the console.

The Card Modules are standard sizes, full and half cards. They slide into the chassis from the rear and are retained via a card retainer bar. Paragraph 9.9 establishes detailed mechanical configurations for the card modules. The card modules are supported when installed, on the side by the chassis card guides, in the front by the motherboard connectors and by the card retainer in the rear.

9.4 CHASSIS CONCEPT

The chassis mechanical design is of unitized sheet metal construction, allowing all subassemblies to be removed without losing structural integrity. The chassis structure is designed to be mounted in a standard cabinet utilizing the vertical "RETMA" rails without the use of side or rear supports. The card guides are an integral part of the chassis. The chassis card guides are on .75 centers to accomodate component heights on card modules of 0.50 inches with the exception of the top card slot (A). The top card slot is spaced 1.1" from the chassis to accomodate the NAKED MINI 16 LSI computer module with a piggyback option card installed. This module has a maximum height of 1.00 inch.



9-4



9.5 MOTHERBOARD ASSEMBLY CONCEPT

The motherboard assembly is a double sided printed circuit board with 5 rows of two 86 pin edge connectors, a 36 pin edge connector and three sets of 50 pin edge connectors.

The five rows of two connectors are used for card module interconnections receiving signal and power. The 36 pin connector is for the DC voltage interface from the power supply to the Card Modules via the motherboard. Three 50 pin connectors are provided for console and expansion chassis interface. J1 is used for the console interface cable while J2 and J3 are interface cables to the expansion chassis.

9.6 FAN HOUSING ASSEMBLY CONCEPT

The fan housing assembly contains two fans, an AC input plug, an AC power switch, a fuse, RFI/EMI filters and the interface connector pigtail (H-P2) for the power supply.

The assembly is a one piece sheet metal chassis.

9.7 POWER SUPPLY ASSEMBLY CONCEPT

The power supply is configured to accomodate all DC components with the AC interfacing at the transformer. The transformer, four filter capacitors and a printed circuit board mount to the chassis. The printed circuit board has a 36-pin connector for DC voltage interface to the motherboard. The printed circuit board contains all the regulator and overvoltage protection circuits. The heat sinks for the DC power circuits are mounted directly to the printed circuit board. The printed circuit board is attached to the chassis via standoffs. The power supply internal wire harness is located between the printed circuit board and the chassis. The power supply chassis is a one piece sheet metal structure with open frame construction allowing easy access to all components.

9.8 CONSOLE ASSEMBLY CONCEPT

The Console assembly consists of the front panel (plastic) and the printed circuit board with an interface cable. The front panel is injection molded plastic. The printed circuit board mounts to the back of the front panel and contains all the components and switches as a complete self contained sub-assembly.



9.9 CARD MODULE PRINTED CIRCUIT BOARD MECHANICAL SPECIFICATION (Figures 9-3 and 9-4)

This paragraph discusses the mechanical design of a printed circuit board which can be installed in an ALPHA 16 LSI computer chassis.

Either full or half printed circuit boards may be used. When half boards are used, two half boards are joined together to form a full board.

All boards use bus bars to distribute power and ground to circuits. The bus bars minimize the ground and power etch runs, leaving more space on the board for signal etched circuit routing. The bus bar design permits etched circuitry to be routed underneath the bus bar with no danger of shorting.

Fiberglass stiffeners are used on all boards to eliminate sag and provide improved structural integrity.

The thickness of the printed circuit board is determined by the motherboard connectors. A typical board is .062 inches thick. The motherboard connector permits variations in thickness ranging from .054 to .071 inches.

All components, stiffeners, bus bars, etc. are mounted on one side of the board. This side of a board is referred to as the "component side", while the other side is referred to as the "solder side". Boards are always installed with the component side up.

Figures 9-3 and 9-4 show the critical dimensions, hole patterns for bus bars and stiffeners and the integrated circuit layout organization for a full and a half board, respectively.

The rear edge of the full board has room for two interface connectors with the exception of the NAKED MINI 16 LSI card. The 1.250 inch dimension from each edge is the area reserved for the card extractors (Part No. 40-06100). The .800 inch dimension at the center is the area reserved for the card retainer. The remaining area along the rear edge is the connector area. The 6.350 inch dimension is the maximum allowable area that the mating connector can occupy. The overall length of a connector cannot exceed this dimension.

The rear edge of a half board has room for only one interface connector. A distance of 1.210 inch must be reserved for a modified card extractor (Part No. 00-00296-00). This leaves 5.080 inches of useable connector area. The 5.080 dimension is the inside contact dimension of the standard 100-pin interface connector.

Half boards must provide for a card extractor at both rear corners although only one is installed depending upon which way the board is strapped to a second half card.



2. PINS 1 AND 7 OF ALL IC PACKAGES TO BE ALIGNED WITH BUS BAR MOUNTING HOLES.
1. NO SILKSCREENING PERMITTED OVER ANY PLATED HOLE.
- NOTES: UNLESS OTHERWISE SPECIFIED

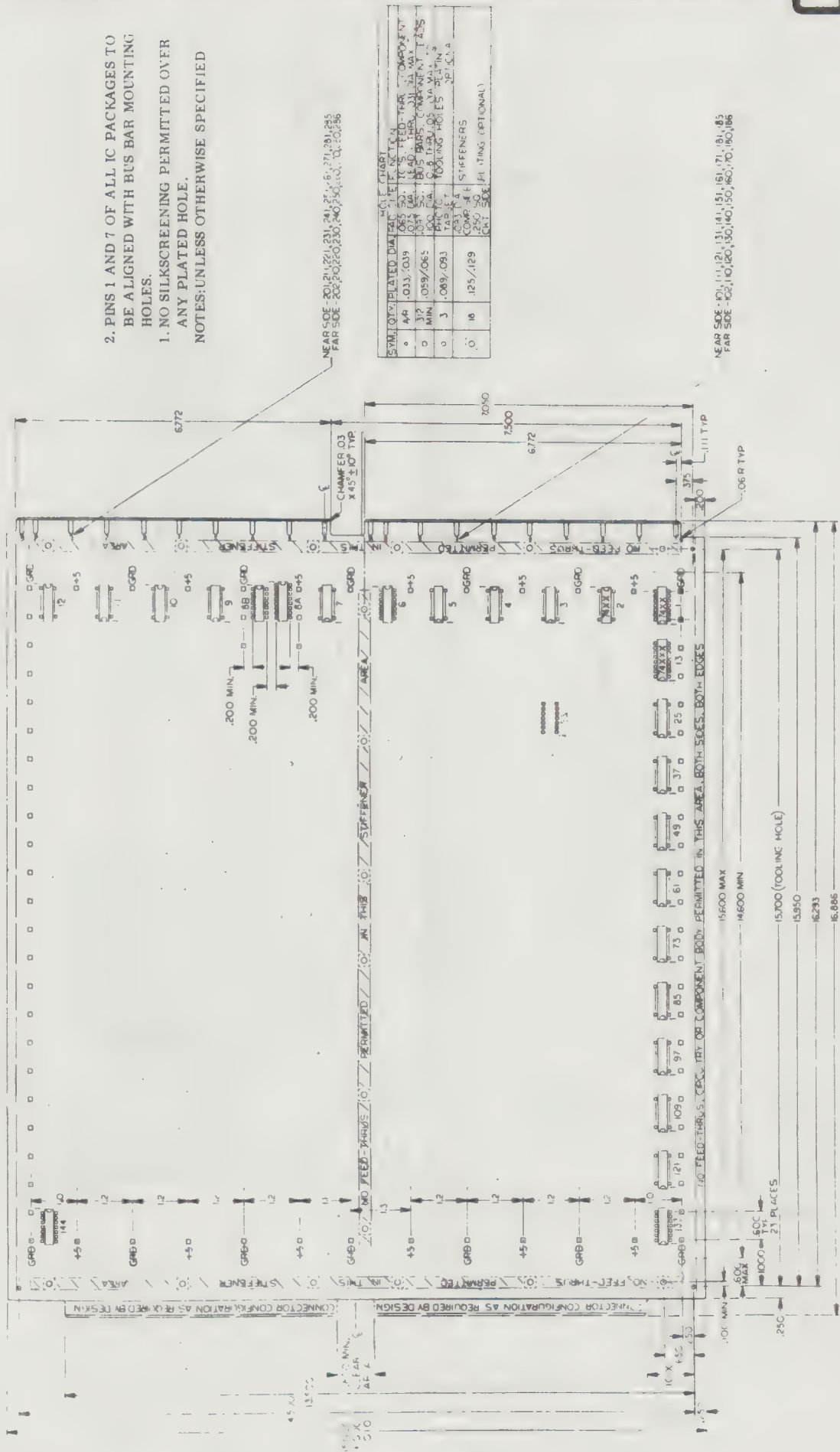


Figure 9-3. Full Board Design Guide



Figure 9-4. Half Board Design Guide



9.10 ENVIRONMENTAL SPECIFICATIONS

The NAKED MINI 16 LSI and ALPHA 16 LSI Computers and the Expansion Chassis will meet the following requirements:

1. Temperature:
 - Operating - 0° C to 55° C
 - Non-Operating - 10° C to 100° C
2. Humidity - 90% no condensation
3. Altitude:
 - Operating - 6500 feet above sea-level
 - Non-Operating - 40,000 feet above sea-level
4. Shock and Vibration - normal bench handling and railroad bumping when in shipping container

The installation requirements for mounting the computer mechanically are established in the outline and mounting drawing, figure 9-2. The enclosure requirements for installation depend on the ventilating system employed such that the thermal requirements are maintained.

There are three installation criteria which provide the minimum cooling conditions allowable for the ALPHA 16 LSI computer and the expansion chassis.

1. Closed Ventilation System
2. Side Ventilation System
3. Top/Bottom Ventilation System

In the closed ventilating system it is assumed the ambient temperature will be maintained by a thermal interface. The minimum size enclosure provides adequate air flow paths for the computer internal fans. See figure 9-5 for installation parameters for the computer using this system.

The side ventilating system establishes the minimum enclosure size and rectangular surface for the minimum size opening. This would provide for a safety guard if necessary. See figure 9-5 for installation parameters for the computer using this cooling system.

The top/bottom ventilating system defines the minimum airflow paths for a cabinet with stacked equipment or an individual console enclosure. See figure 9-5 for installation parameters of the computer using this system.

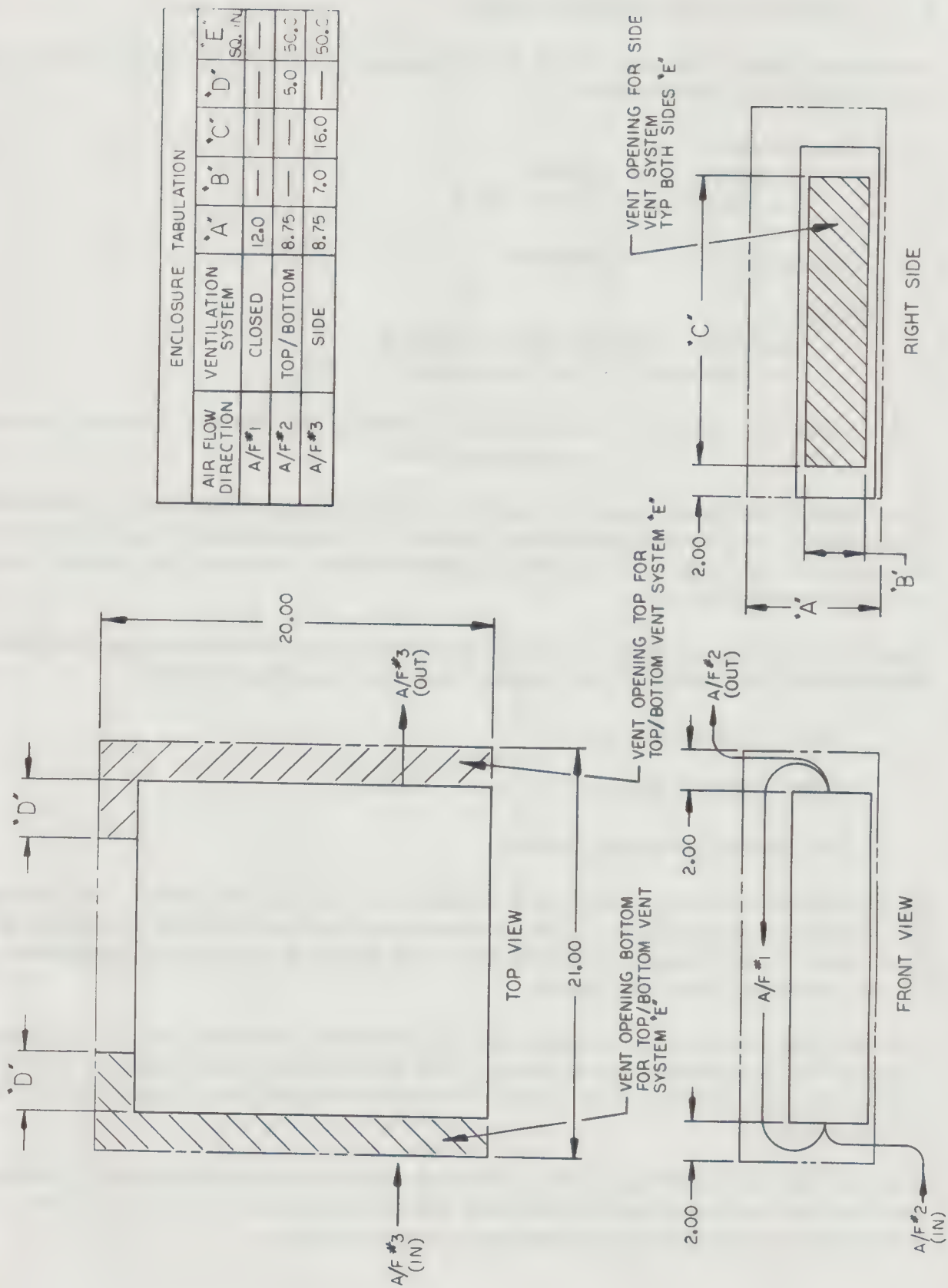


Figure 9-5. ALPHA 16 LSI Ventilation Systems



Appendix A

INSTRUCTION SET IN NUMERICAL ORDER

This appendix contains the ALPHA LSI instruction set in numerical machine code order. Instructions with variable fields (D,K, etc.) have been appended with an asterisk (*).

<u>Instruction Code In Hex</u>	<u>Instruction Mnemonic</u>	<u>Description</u>
0000	NOP	NO OPERATION
0008	XRM	X REGISTER TO MINUS ONE
0010	ARM	A REGISTER TO MINUS ONE
0018	AXM	A AND X REGISTERS TO MINUS ONE
0030	TXA	TRANSFER X TO A
0048	TAX	TRANSFER A TO X
0068	ANX	AND OF A AND X TO X
0070	ANA	AND OF A AND X TO A
00A8	DXR	DECREMENT X REGISTER
00B0	DXA	DECREMENT X TO A
00C8	DAX	DECREMENT A TO X
00D0	DAR	DECREMENT A REGISTER
0108	ZXR	ZERO X REGISTER
0110	ZAR	ZERO A REGISTER
0118	ZAX	ZERO A AND X
0128	IXR	INCREMENT X REGISTER
0130	IXA	INCREMENT X TO A
0148	IAX	INCREMENT A TO X



<u>Instruction Code In Hex</u>	<u>Instruction Mnemonic</u>	<u>Description</u>
0150	IAR	INCREMENT A REGISTER
0208	CAX	COMPLEMENT OF A TO X
0210	CAR	COMPLEMENT A REGISTER
0308	NAX	NEGATE A TO X
0310	NAR	NEGATE A REGISTER
0350	ARP	A REGISTER TO PLUS ONE
0358	AXP	A AND X REGISTERS TO PLUS ONE
0408	CXR	COMPLEMENT X REGISTER
0410	CXA	COMPLEMENT OF X TO A
0508	NXR	NEGATE X REGISTER
0510	NXA	NEGATE X TO A
0528	XRP	X REGISTER TO PLUS ONE
0608	NRX	NOR OF A AND X TO X
0610	NRA	NOR OF A AND X TO A
0800	HLT	HALT
0800	STOP*	HALT WITH OPERAND
0A00	EIN	ENABLE INTERRUPTS
0C00	DIN	DISABLE INTERRUPTS
0E00	SBM	SET BYTE MODE
1028	ALX*	ARITHMETIC SHIFT X LEFT
1050	ALA*	ARITHMETIC SHIFT A LEFT
10A8	ARX*	ARITHMETIC SHIFT X RIGHT
10D0	ARA*	ARITHMETIC SHIFT A RIGHT



<u>Instruction Code in Hex</u>	<u>Instruction Mnemonic</u>	<u>Description</u>
1128	RLX*	ROTATE X LEFT WITH OVERFLOW
1150	RLA*	ROTATE A LEFT WITH OVERFLOW
11A8	RRX*	ROTATE X RIGHT WITH OVERFLOW
11D0	RRA*	ROTATE A RIGHT WITH OVERFLOW
1200	ROV	RESET OVERFLOW
1320	SXO	SIGN OF X TO OVERFLOW
1328	LLX*	LOGICAL SHIFT X LEFT
1340	SAO	SIGN OF A TO OVERFLOW
1350	LLA*	LOGICAL SHIFT A LEFT
13A0	LXO	LSB OF X TO OVERFLOW
13A8	LRX*	LOGICAL SHIFT X RIGHT
13C0	LAO	LSB OF A TO OVERFLOW
13D0	LRA*	LOGICAL SHIFT A RIGHT
1400	SOV	SET OVERFLOW
1600	COV	COMPLEMENT OVERFLOW
1900	LRL*	LONG ROTATE LEFT
1940	NRM	NORMALIZE A AND X
1960	MPY	MULTIPLY AND ADD
1970	DVD	DIVIDE
1980	LRR*	LONG ROTATE RIGHT
1B00	LLL*	LONG LOGICAL SHIFT LEFT
1B80	LLR*	LONG LOGICAL SHIFT RIGHT



<u>Instruction Code in Hex</u>	<u>Instruction Mnemonic</u>	<u>Description</u>
2080-3F80 Forward 20C0-3FC0 Backward	JOC*	JUMP ON CONDITION
2080 Forward 20C0 Backward	JAM*	JUMP IF A MINUS
2100 Forward 2140 Backward	JAZ*	JUMP IF A ZERO
2180 Forward 21C0 Backward	JAL*	JUMP IF A LESS THAN ONE
2200 Forward 2240 Backward	JOS*	JUMP IF OVERFLOW SET
2400 Forward 2440 Backward	JSR*	JUMP IF SENSE SWITCH RESET
2800 Forward 2840 Backward	JXZ*	JUMP IF X ZERO
3080 Forward 30C0 Backward	JAP*	JUMP IF A POSITIVE
3100 Forward 3140 Backward	JAN*	JUMP IF A NOT ZERO
3180 Forward 31C0 Backward	JAG*	JUMP IF A GREATER THAN ZERO
3200 Forward 3240 Backward	JOR*	JUMP IF OVERFLOW RESET
3400 Forward 3440 Backward	JSS*	JUMP IF SENSE SWITCH SET
3800 Forward 3840 Backward	JXN*	JUMP IF X NOT ZERO
4000	SEL*	SELECT FUNCTION
4002	PFE	POWER FAIL ENABLE
4003	PFD	POWER FAIL DISABLE



<u>Instruction Code in Hex</u>	<u>Instruction Mnemonic</u>	<u>Description</u>
4005	CIE	CONSOLE INTERRUPT ENABLE
4006	CID	CONSOLE INTERRUPT DISABLE
4007	TRP	TRAP
4400	SEA*	SELECT AND PRESENT A
4404	OCA	OUTPUT A TO CONSOLE REGISTER
4600	SEX*	SELECT AND PRESENT X
4604	OCX	OUTPUT X TO CONSOLE REGISTER
4800	SSN*	SENSE AND SKIP ON NO RESPONSE
4900	SEN*	SENSE AND SKIP ON RESPONSE
5000	AIN*	AUTOMATIC INPUT WORD TO MEMORY
5400	AIB*	AUTOMATIC INPUT BYTE TO MEMORY
5800	INA*	INPUT TO A REGISTER
5800	SIA	STATUS INPUT TO A
5801	ISA	INPUT SENSE REGISTER TO A
5804	ICA	INPUT CONSOLE REGISTER TO A
5900	RDA*	READ WORD TO A REGISTER
5A00	INX*	INPUT TO X REGISTER
5A01	ISX	INPUT SENSE REGISTER TO X
5A04	ICX	INPUT CONSOLE REGISTER TO X
5B00	RDX*	READ WORD TO X REGISTER
5C00	INAM*	INPUT TO A REGISTER MASKED
5D00	RDAM*	READ WORD TO A REGISTER MASKED
5E00	INXM*	INPUT TO X REGISTER MASKED



<u>Instruction Code in Hex</u>	<u>Instruction Mnemonic</u>	<u>Description</u>
5F00	RDXM*	READ WORD TO X REGISTER MASKED
6000	AOT*	AUTOMATIC OUTPUT WORD FROM MEMORY
6400	AOB*	AUTOMATIC OUTPUT BYTE FROM MEMORY
6800	OTZ*	OUTPUT ZERO
6800	SIN*	STATUS INHIBIT
6900	WRZ*	WRITE ZERO
6C00	OTA*	OUTPUT A REGISTER
6C00	SOA	STATUS OUTPUT FROM A
6D00	WRA*	WRITE FROM A REGISTER
6E00	OTX*	OUTPUT X REGISTER
6E00	SOX	STATUS OUTPUT FROM X
6F00	WRX*	WRITE FROM X REGISTER
7100	BIN*	BLOCK IN
7500	BOT*	BLOCK OUT
7800	IBA*	INPUT BYTE TO A REGISTER
7900	RBA*	READ BYTE TO A REGISTER
7A00	IBX*	INPUT BYTE TO X REGISTER
7B00	RBX*	READ BYTE TO X REGISTER
7C00	IBAM*	INPUT BYTE TO A REGISTER MASKED
7D00	RBAM*	READ BYTE TO A REGISTER MASKED
7E00	IBXM*	INPUT BYTE TO X REGISTER MASKED
7F00	RBXM*	READ BYTE TO X REGISTER MASKED



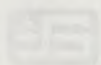
<u>Instruction Code in Hex</u>	<u>Instruction Mnemonic</u>	<u>Description</u>
8000	AND*	AND TO A
8000	ANDB*	AND BYTE TO A
8800	ADD*	ADD TO A
8800	ADDB*	ADD BYTE TO A
9000	SUB*	SUBTRACT FROM A
9000	SUBB*	SUBTRACT BYTE FROM A
9800	STA*	STORE A
9800	STAB*	STORE A BYTE
A000	IOR*	INCLUSIVE OR TO A
A000	IORB*	INCLUSIVE OR BYTE TO A
A800	XOR*	EXCLUSIVE OR TO A
A800	XORB*	EXCLUSIVE OR BYTE TO A
B000	LDA*	LOAD A
B000	LDAB*	LOAD A BYTE
B800	EMA*	EXCHANGE MEMORY AND A
B800	EMAB*	EXCHANGE MEMORY BYTE AND A
C000	CAI*	COMPARE TO A IMMEDIATE
C100	CXI*	COMPARE TO X IMMEDIATE
C200	AXI*	ADD TO X IMMEDIATE
C300	SXI*	SUBTRACT FROM X IMMEDIATE
C400	LXP*	LOAD X POSITIVE IMMEDIATE
C500	LXM*	LOAD X MINUS IMMEDIATE



<u>Instruction Code in Hex</u>	<u>Instruction Mnemonic</u>	<u>Description</u>
C600	LAP*	LOAD A POSITIVE IMMEDIATE
C700	LAM*	LOAD A MINUS IMMEDIATE
CD00	SCM*	SCAN MEMORY
D000	CMS*	COMPARE AND SKIP IF HIGH OR EQUAL
D000	CMSB*	COMPARE BYTE AND SKIP IF HIGH OR EQUAL
D800	IMS*	INCREMENT MEMORY AND SKIP ON ZERO RESULT
E000	LDX*	LOAD X
E000	LDXB*	LOAD X BYTE
E800	STX*	STORE X
E800	STXB	STORE X BYTE
F000	JMP*	JUMP UNCONDITIONAL
F600	WAIT	WAIT FOR INTERRUPTS
F800	JST*	JUMP AND STORE



COMPUTER AUTOMATION, INC.
the MAKED MINI company



COMPUTER AUTOMATION

the MAKED MINI company

CAI Limited
2nd Floor, West
10, Parkway Court, Wexham, Wexham, Wexham, Wexham
Wexham, Wexham, Wexham, Wexham, Wexham, Wexham, Wexham, Wexham



COMPUTER AUTOMATION, INC.
the NAKED MINI company



COMPUTER AUTOMATION
the NAKED MINI company

CAI Limited,
95a High Street,
Rickmansworth, Herts, WD3 1RB
Tel: Rickmansworth 79704/7

Telex: 922654